2016

Enabling Machine Science through Distributed Human Computing

Mark David Wagy
University of Vermont

Enabling Machine Science Through Distributed Human Computing

A Dissertation Presented

by

Mark Wagy

to

The Faculty of the Graduate College

of

The University of Vermont

In Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
Specializing in Computer Science

October, 2016

Defense Date: June 2nd, 2016
Dissertation Examination Committee:

Josh Bongard, Ph.D., Advisor
Paul Hines, Ph.D.
Robert Snapp, Ph.D.
Elizabeth Pope, Ph.D., Chairperson
Cynthia J. Forehand, Ph.D., Dean of Graduate College

Abstract

Distributed human computing techniques have been shown to be effective ways of accessing the problem-solving capabilities of a large group of anonymous individuals over the World Wide Web. They have been successfully applied to such diverse domains as computer security, biology and astronomy. The success of distributed human computing in various domains suggests that it can be utilized for complex collaborative problem solving. Thus it could be used for "machine science": utilizing machines to facilitate the vetting of disparate human hypotheses for solving scientific and engineering problems.

In this thesis, we show that machine science is possible through distributed human computing methods for some tasks. By enabling anonymous individuals to collaborate in a way that parallels the scientific method – suggesting hypotheses, testing and then communicating them for vetting by other participants – we demonstrate that a crowd can together define robot control strategies, design robot morphologies capable of fast-forward locomotion and contribute features to machine learning models for residential electric energy usage. We also introduce a new methodology for empowering a fully automated robot design system by seeding it with intuitions distilled from the crowd.

Our findings suggest that increasingly large, diverse and complex collaborations that combine people and machines in the right way may enable problem solving in a wide range of fields.

Citations

Material from this dissertation has been published in the following form:

Wagy, Mark, and Bongard, Josh C.. (2014). "Collective design of robot locomotion." ALIFE 14: The Fourteenth Conference on the Synthesis and Simulation of Living Systems. Vol. 14.

AND

Wagy, Mark D., and Bongard, Josh C.. (2015). "Combining Computational and Social Effort for Collaborative Problem Solving." PloS one 10.11: e0142524.

AND

Wagy, Mark D., and Bongard, Josh C.. (2015). "Crowdseeding robot design." Proceedings of the Companion Publication of the 2015 on Genetic and Evolutionary Computation Conference. ACM. (Short paper / extended abstract format)

AND

Wagy, Mark D., and Bongard, Josh C.. (2015). "Crowdseeding: a novel approach for designing bioinspired machines." Biomimetic and Biohybrid Systems. Springer International Publishing. 293-303.

Material from this thesis has been accepted for publication in "ALIFE 16: The Sixteenth Conference on the Synthesis and Simulation of Living Systems. Vol. 16." in the following form:

Wagy, Mark D., and Bongard, Josh C.. (2016). "Social contribution in the design of Adaptive Machines on the Web." ALIFE 16: The Sixteenth Conference on the Synthesis and Simulation of Living Systems. Vol. 16.

Material from this dissertation has been submitted for publication to IEEE Transactions on Smart Grid on May 9, 2015 in the following form:

Wagy, Mark D., Bongard, Josh C., Bagrow, James P., Hines, Paul D.H.. (2016) "Crowdsourcing Predictors of Residential Electric Energy Usage." IEEE Transactions on Smart Grid.

Acknowledgements

First and foremost, I want to express deep thanks to my Ph.D. adviser, Josh Bongard, for his mentorship and guidance. I would also like to thank the members of my committee: Robert Snapp, Lizzy Pope and in particular Paul Hines for his mentorship, instruction and collaboration throughout all years of my Ph.D. Additionally, I would like to thank Maggie Eppstein for helpful discussion and guidance throughout the first year of my Ph.D.

I would like to extend a special thanks to my parents for their encouragement and guidance. I am also very grateful to the support of my parents-in-law, Giuseppe and Giuseppa Porcelli – in particular Giuseppa, who came to the United States to live with us and take care of our son for many of the months after he was born so that I could work on research. Grazie mille.

These past few years have been enlivened and enrichened by fellow Ph.D. students-turned-friends. In particular (alphabetical), Christopher (or is it Chris?), Curtis, Dan, Emily and Tom. Oh, and Bob.

And finally to my wife, Emanuela, who has been patient and supportive throughout the Ph.D experience.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Through the use of distributed human computing we have an unprecedented ability to distribute tasks to a large group of anonymous individuals over the World Wide Web ("the Web"). The Web gives us access to this group of individuals without regard to their geographic location or background [1,2]. Thus a crowd can contribute to problems in a way that was previously only possible through direct interaction by a physically co-located group of participants, or through costly and slow mail or telephone coordination. With such an ability for a crowd to contribute to computing tasks comes novel applications for the incorporation of both individual and social contribution of human agents in distributed computing systems.

In this thesis, we demonstrate the capacity for utilizing distributed human computing methods to contribute to solving scientific and engineering problems. We hypothesize that it is possible to enable machine science through collective intelligence and distributed human computing methods. The crowd can do more than simply see patterns in data, provide computation and provide data: they can individually generate and collectively vet hypotheses.

Collective Intelligence

A collective is a group of agents that work together, without regard to the specific social dynamics that take place within the group. In broad terms, these agents can be biological or artificial, human or animal. Previous literature on collective decision making has suggested that collective decision making is similar to cognitive processes that take place within neural activity in the brain [3]. Thus we can approach such social systems as a novel form of intelligence-as-it-could-be [4] in which a collective works together towards a common goal in the same way that we consider processes in the brain synergize to demonstrate intelligent behavior, such as solving problems. We adopt such a view of collective intelligence in this thesis.

By using the term "collective", we can refer to a group of agents that are in competition, a group in cooperation or some mixture of cooperation and competition. Adopting this definition of collective efforts allows us to refer to the group in aggregate rather than making claims about what specific interactions take place between members of the group. Any reference to a team, group or organization in this thesis can be read as synonymous with this definition of a collective. Thus we approach a collective as a black-box modular component within the context of a larger computing system. This view allows us to swap a module containing a human collective with a group of isolated human agents and compare their performance.

Collective intelligence is the ability of a group of people to accomplish a task as a group rather than perform that task individually. When we refer to *intelligence* in this thesis, we are invoking a reference to the weak definition of artificial intelligence: that an intelligence system act rationally with respect to solving a particular problem

rather than measuring intelligence with respect to the generals abilities of a human. It is with this understanding of intelligence that we can consider a distributed computing system to manifest intelligent behavior with respect to its ability to solve problems. In keeping with this 'weak' form of artificial intelligence, we can use the definition of collective intelligence to refer to problem solving behavior that results in a behavior that seeks to maximize some objective, such as minimizing an error. Thus a group of humans and machines may not exhibit human-like intelligence when viewed as a collective, but if they are behaving rationally with respect to an objective they are exhibiting intelligent behavior.

Distributed Human Computing Methods

We refer to the means by which we can achieve (human-based) collective intelligence as distributed human computing [5]. We proceed according to the definitions of distributed human computing methodologies as outlined in previous work [6] and briefly summarize them here.

Distributed human computing (synonymous with collective intelligence, but distinguished by referring to a collective of humans rather than other animals) can be subdivided into four overlapping categories: crowdsourcing, human computation, social computing and data mining.

Crowdsourcing refers to the use of non-experts for work that would otherwise have been performed by experts. The term was derived in analogy with the practice of *outsourcing* [7] in which companies hire geographically diverse workers to complete tasks that would traditionally have been completed by geographically co-located workers.

Most crowdsourcing tasks involve the recruitment of mainly non-expert contributors through the use of technology; specifically via the Web. Crowd contributions have been shown to be similar in quality to that of experts for some tasks [8].

Human computation [9] refers to the use of humans for tasks that cannot yet be performed adequately by machines. Commonly these tasks incorporate the superior pattern recognition capabilities of humans over machines [10, 11].

Social Computing [12,13] is exemplified by social networking sites such as Facebook or LinkedIn in which users interact socially through the use of technology. Thus social computing refers to the use of technology to enhance social interaction or present social interactive outlets that were not available without the use of technological means. However, the social activity has no specific task or problem that is being solved as a collective. The work in this thesis utilizes social computation, but for the purpose of solving specific problems.

Previously defined taxonomies of human computation [6] have also referred to *Data Mining* as a fourth form of distributed human computing. This is to accommodate the fact that many datasets are the result of collective human contributions, and that data mining methods require the combined efforts of humans to build inferences from the data source. The resulting inferences are therefore the result of distributed human computing efforts.

In this work, we focus the area of intersection between these distributed computation methods. We utilize the pattern recognition capabilities of humans to detect patterns, and therefore draw upon human computation literatures. We utilize methods from crowdsourcing: we can access a large group of diverse workers with diverse capabilities through the Web. And we rely upon social computing in that we draw

upon the collective work of individuals through machine-enabled interaction over the Web.

Distributed human computing (DHC) methods have been demonstrated to be effective at utilizing a group of loosely connected web users for a wide range of domain applications: folding protein structures [14], galaxy classification [15], computer security [10], and mapping the brain [16]. By reformulating problems as games, participants in crowdsourcing tasks have demonstrated the capacity to participate in complex problem solving tasks when given proper incentive [17]. However, the vast majority of crowdsourcing examples have focused on problems with easily separable solutions, which are amenable to divide-and-conquer computing strategies: a task or set of tasks are broadcast to workers, completed independently and returned to the requester. Thus, crowdsourcing methods have focused on distributing work to isolated individuals in a hub-and-spoke [18] fashion rather than utilizing the collective intelligence of participants in the crowdsourced task by encouraging horizontal interaction to formulate solution hypotheses collaboratively.

DHC techniques have exploited the capabilities of machines – those of rapid calculation, communication and interactivity – to enable disparate, geographically diverse crowds to solve problems effectively through crowdsourcing. But it remains to be seen in what ways machines can enable the crowd to produce better ideas as a collective. Previous efforts have integrated human abilities into machine-human interactive systems, particularly through interactive evolutionary algorithms [19]. Individual human participants have contributed to fitness evaluation of art pieces [20, 21], robot behavior [22–24], developing artificial organisms [25] and engineering artifacts [26]. Not only have humans been used for evaluation in genetic algorithms, but they have been used

in creating new prospective solutions to be vetted in evolutionary algorithms [27]. Outside of interactive evolutionary algorithms, studies have crowdsourced the discovery of common sense rules to build machine intelligence models [28–31]. However, in these studies there is little to no interaction between the participants in the crowdsourced task. And there has been no clear demonstration that collaborative efforts at hypothesis formulation outperform independent efforts. The focus has rather been on whether the crowd can be used to generate a viable solution.

The hub-and-spoke approach to crowdsourcing typically relies on a centrally-located 'requester' to ask workers to complete simple tasks. The workers are asked to perform tasks without the benefit of collaboration. However, there are crowdsourcing examples that utilize the capabilities of the crowd as a collective entity. Wikipedia, an encyclopedia that relies only on user-contributed entries, has equaled the quantity and quality of traditionally constructed encyclopedias [32] through user collaboration. However, Wikipedia is a commercial endeavor and – although a source of data and the subject of research – has no control to which it can be compared to demonstrate the effectiveness of the collaborative efforts of the crowd; and no explicit problem is being solved by the crowd. Therefore, despite Wikipedia being an instance of collaborative crowdsourcing, there is no way in which we can consider the participants to be participating in hypothesis suggestion as it is in an open-ended crowdsourced activity. The discovery of novel proofs to mathematical theorems has been demonstrated through collaborative wiki and blogs [33]. However, in this example, problem solving was not crowdsourced: The discovery of novel ways in which a mathematical proof could proceed was facilitated mainly by experts. And in all crowdsourcing examples, there is a paucity of investigation into the capabilities of machines versus the contributions

of humans.

Machine Science and Distributed Human Computing

Machine science is a method to automate aspects of the scientific process [34]. One way to do this is to find connections in public data stores between human-generated hypotheses [35,36]. In this thesis, we demonstrate that machine science over the Web is possible by providing a set of anonymous participants over the Web the capability of testing, communicating and vetting hypotheses.

Through machine science we can use machine capabilities to combine unrelated hypotheses and concepts formulated under isolating circumstances, such as differences in language, culture, or geographic location. By crowdsourcing machine science, we have access a vast number of diverse individuals with varying perspectives. Using distributed human computing methods for machine science can help to overcome any such inabilities to communicate on a conceptual or practical level. Evidence suggests that a group of diverse individuals can outperform a group of high-ability problem-solvers – that 'diversity trumps ability' [37]. Thus enabling machine science through DHC, we can combine diverse perspectives as well as harvest the efforts of non-experts for collaborative problem solving.

However, we cannot assume that collective efforts are necessarily constructive. In fact there are well-known group pathologies that can arise in team efforts. Pathologies such as groupthink [38] can lead to sub-optimal ideas that dominate group efforts, or the Ringelmann effect [39] can lead to reduced productivity as group size increases.

In the experiments that follow, we show that if such group pathologies were present, they were overshadowed by group synergies.

Summary

In order to show that machine science can be enabled through distributed human computing, we need to demonstrate that anonymous, non-experts can work together over the Web to solve difficult problems. We must show that people working over the web do not fall prey to group pathologies that would prevent them from working together despite not being physically co-located. We must also demonstrate that hypotheses formed under social influence are indeed better than if they were formed in isolation. Furthermore, we must show that incorporating such socially constructed hypotheses is useful over simply relying upon machines to do the same. And we can strengthen the claim that machine science can be enabled by distributed human computing by demonstrating that it can be successful in multiple domains.

In this thesis, we show that the crowd is able to perform tasks once deemed only possible by a small group of engineers or scientists. The crowd is able to contribute to solving problems by suggesting hypotheses, testing those hypotheses and communicating and vetting them with the aid of machine-to-machine communication, human-computer interaction and machine learning algorithms.

We show that non-expert humans and computers can be combined to solve several problems in robotics. We first show that humans can be tasked with designing controllers for autonomous robots, while the computers seek to optimize the controllers designed by humans. In this application we describe the implementation of

the first-ever web-based crowdsourced robotics design platform (Chapter 2). Users can test hypotheses for robot movement using a Web-embedded physics simulation. These hypotheses are then communicated with other participants to be vetted. We compare a control group in which users are isolated from the hypotheses of others to an experimental group in which users can draw upon hypotheses suggested by others. In doing so, we can determine whether those who are able to share hypotheses do so constructively or destructively. Or whether the sharing of hypotheses leads to no advantage or disadvantage at all.

We then describe a system that allows an anonymous group of workers to design robot bodies over the Web. They are presented with the objective of hypothesizing which robot bodies are able to rapidly walk over flat ground. They are presented with hypotheses tested by other contributors for vetting in the same way as in the robot control experiment. We demonstrate that this anonymous collective of non-experts are better able to design locomoting robot bodies than a state-of-the-art automated algorithm (Chapter 3). Furthermore, we show that communicating hypotheses for vetting in this way results in superior performing robots than if we were to isolate participants from others' designs.

When compared with fully automated algorithms for robot design, we find that human contributors favored designs that were symmetric, a quality that might be the result of exposure to walking animals in Nature (Chapter 3). We also find a latent, measurable quantity that is more prevalent in hypotheses contributed by those who are able to share versus those that work alone; and that this quantity correlates to the objective outcome of the design problem, which is fast forward locomotion (Chapter 4).

We show that this process of generating and vetting hypotheses through social interaction is useful outside of the domain of robotics. We demonstrate that non-experts can contribute hypotheses in the form of questions to be used in a predictive model of electric energy usage (Chapter 5).

We also develop a new technique to distill crowd preferences and then incorporate them into fully automated algorithms for robot designs, that we have named 'crowd-seeding'. By incorporating features that were discovered both manually (Chapter 6) and in an automated fashion (Chapter 7), we demonstrate that we are able to design better robots when we incorporate human feedback.

Thus we demonstrate that, via the testing and vetting of hypotheses over the Web, collective groups do not necessarily fall prey to group pathologies. Participants were able to devote computational effort and design effort to designing both robot control strategies and robot bodies in a way that outperformed isolated individuals and an automated robot design algorithm that is considered state-of-the-art. And we show that there exists a constructive quality in social hypothesis generation that correlates with the objective of a problem, namely that of building robot bodies capable of rapid locomotion. Individuals are not only able to constructively contribute hypotheses via distributed human computing in the robotics domain, but they are capable of contributing useful hypotheses to the domain of residential energy usage. And if we distill crowd hypotheses via crowdseeding, we can augment the performance of automated robot design algorithms. Therefore, we show in the following experiments that the ability of the crowd to constructively contribute hypotheses as a collective via testing and vetting of their hypotheses in a Web interface does indeed enable machine science.

Chapter 2

Crowd-Contributed Robot Control Strategies*

Abstract

It has been shown that the collective action of non-experts can compete favorably with an individual expert or an optimization method on a given problem. However, the best method for organizing collective problem solving is still an open question. Using the domain of robotics, we examine whether cooperative search for design strategies is superior to individual search. We use a web-based robot simulation to determine whether groups of human users can leverage design intuition from others to focus search on relevant parts of a complex design space. We show that individuals that work cooperatively with the aid of a simple optimization algorithm are better able to improve the design of robot locomotion than if they were to work individually with the aid of the optimization algorithm. This result suggests that groups of designers may more effectively work in tandem with optimization algorithms than individuals working in isolation.

*Appeared as Wagy, Mark, and Josh Bongard. "Collective design of robot locomotion". ALIFE 14: The Fourteenth Conference on the Synthesis and Simulation of Living Systems. Vol. 14, 2014.

2.1 Introduction

There is a long history of humans interacting with computer systems to design software agents. In one of the most famous examples of interactive design of software agents, human users interacted with Dawkins' *Blind Watchmaker* to generate "biomorphs": increasingly lifelike beings that resulted from a process of interactive selection [25]. However, there is little basis for groups of humans interacting to design software agents collaboratively.

We know that positive feedback within a collective can spread through the group and result in wider perceptual abilities than that of the individual [3]. These types of positive feedback or autocatalytic phenomena may result in behavior that outperforms what any single agent in a population could achieve. Anecdotally, we know that teamwork in an organization can result in creative approaches to problem solving and that the intelligence of a group cannot be predicted by the average intelligence of its individual members [40]. So might a group working with an optimization algorithm collectively arrive at better performing designs than an individual would on their own? With this question in mind, we use a web-based platform for robot and behavior optimization to determine whether groups of human users can leverage design intuition from others to focus search on promising regions of a complex design space.

In order for users to be able to easily communicate designs, we have adoped a means of communicating through a language of graphical symbols, which in some sense acts as a "visual programming language". Visual programming structures can result in better "closeness of mapping" than textual representations [41]. That is,

visual languages map programming abstractions more directly to the domain that they are modeling. We exploited the intrinsic pattern recognition capabilities of humans in the experiments reported here by developing a visual language for robot configuration. Users 'wire' subsets of a robot's joints together using colored lines, which constrains those subsets to move in phase with one another. An underlying evolutionary optimization algorithm then optimizes the movements of the joints to produce locomotion. Subsequent visitors to the site can then ascertain promising areas of the search space by visually inspecting the wiring patterns created by others and contribute their own computer's time to those regions, or create new designs of their own.

Three main literatures informed the design of our experiment: interactive evolutionary algorithms, visual design languages, and collaborative problem solving.

2.1.1 Interactive Evolutionary Algorithms

A group of non-experts, even children, have been found capable of training robots to achieve complex movement. [22] employed child subjects in a programming-free, interactive evolutionary robotics experiment to train artificial neural networks to produce interesting robot behaviors such as obstacle avoidance and line and wall following. [24] used multiplayer online games featuring simulated robot agents to train a Case-Based Reasoning system. The robots then interacted with humans in a physical environment similar to the web-based simulation in which they were trained. However, the focus of most crowdsourcing robotics studies has been on using individual human users to train robots. Similarly, in studies involving interactive evolutionary robotics, problem solving primarily involved a single individual dedicated to each

training instance [23], [22], [42]. There was a one-to-one relationship between the human interacting with a robot control algorithm, rather than a group of individuals collectively working toward developing better robot control. Typically the individual directly evaluated fitness of computer-generated designs. [26] gives a broad overview of how interactive evolutionary computation has been used for engineering and creative domains, from the design of hearing aids to evaluation of 3D CG images. The designs tended to be evaluated by a single human interacting individually with an evolutionary computing process. [43] demonstrated the use of interactive evolutionary algorithms in robot control to overcome local optima through demonstration. Also, [44] used human users to train a user model to avoid local optima in an interactive evolutionary algorithm for the control of robot movement. That work built on the user modeling approach to interactive evolutionary robotics pioneered in [45]. Recent work on exploring the possibility of tapping into user creativity using interactive evolutionary algorithms can be seen in [46] and [47].

2.1.2 Visual Design Languages

A substantial challenge in collaborative design is effectively communicating complex concepts between members of a group. In order to allow members of a group to build upon prior knowledge and thus improve past designs, key concepts in the problem domain must be communicated through some form of *working memory*. [48] suggested that the use of a diagram is a means for storing problem states in a working memory. With this idea in mind, we developed a simple visual language of symbols for robot joint configuration. Users working together use these symbols as a blackboard

system [49] to "exchange" ideas and improve the robot design.

Simplification of robot control by defining domain-specific languages is common in the robotics literature. For example, [50] define a declarative language for controlling robot behavior. Their framework is also focused on defining robust robot control rather than performing a single task such as fast locomotion. [51], [52] and [53] all successfully employed visual programming languages for robot control. However, in the present work we are using symbols as a means of indirectly constraining the evolution of high-level robot behavior, rather than programming specific responses of the robot to its environment. It is not the intent of our study to be able to train a robot to exhibit complex behavior with our symbolic cues. Rather, the symbols that represent different classes of robot movement act as a means by which to communicate possible unexplored parts of the evolutionary search space to others. Though in both cases, the importance of the direct mapping of visual indicators to the movement of robots is noted as an important benefit of visual over textual representations of a complex problem such as robot control.

2.1.3 Collaborative Problem Solving

Much of crowdsourcing work is focused on taking a large problem and dividing it into independent "microtasks" to be implemented by independent web-based workers. However, one of the most visible platforms for crowdsourcing is a collaborative one. The crowdsourced online encyclopedia Wikipedia [†] is one of the best examples of

[†]www.wikipedia.org

the success of collaborative human computing. Collaborative, networked design tasks have also shown promise in past literature. [54] deployed a virtual reality framework in which the users of the system directly work with each other on product design. This contrasts with the present work in which individuals in the experimental group only indirectly collaborate with each other through the sharing of past designs asynchronously, but without the intent to directly work with each other. Users implicitly work towards a common goal, without directly working together [55], similar to the approach described in [11]. Collaborative crowdsourcing of design tasks with an evolutionary computing backbone has also been explored in past work. [56] crowdsourced selection and recombination in an evolutionary process of designing alarm clocks. They shared drawings of alarm clocks among Mechanical Turk workers to iteratively vet more creative and practical designs through a process of drawing combination. They show a significant improvement in the practicality and creativity of designs at the third generation versus the initial designs. The *Picbreeder* system developed by [57] employed a web-based collaborative system to create realistic images using the NEAT algorithm [58]. In this paper we investigate whether such collaborative dynamics can be brought to bear on robot design.

2.2 Methodology

We deployed a web-based platform (see Figure 7.1 for a screenshot) in which users "design" different robots, and then an evolutionary optimization algorithm evolves behaviors for them. Users were partitioned into control and experimental groups:

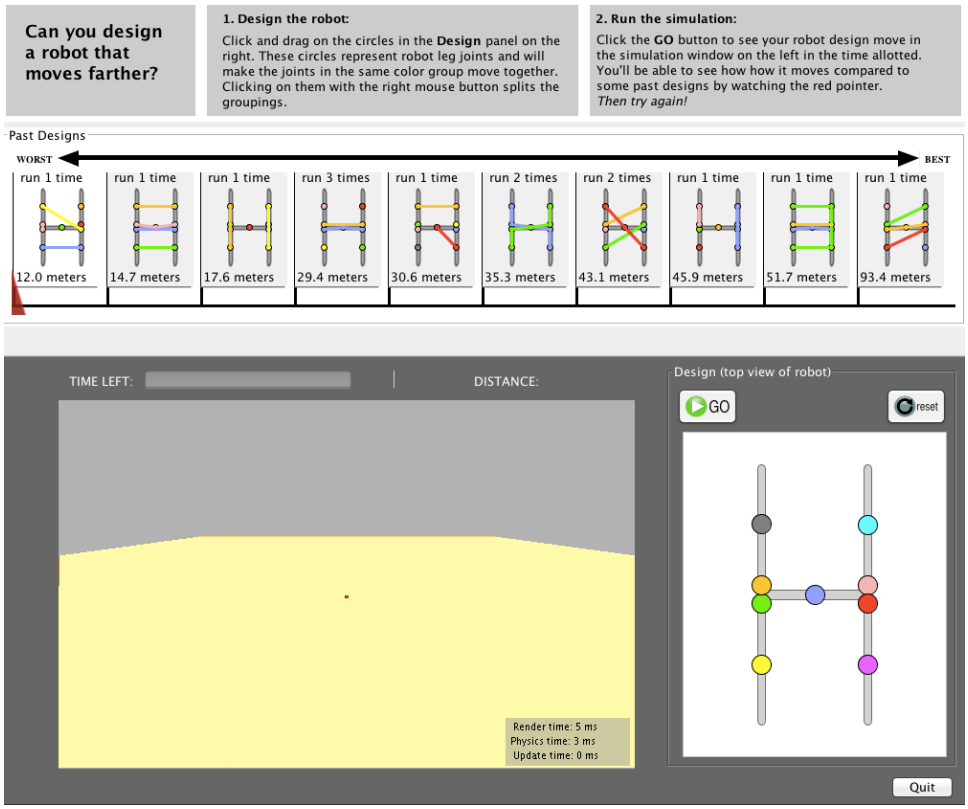

Figure 2.1: ***Screenshot of web-based design tool.*** *A description of how to use the tool is on the top with an arrow indicating that the ordering of designs is from best to worst. The red arrow below the designs indicates in real-time how the current robot compares to the historical designs displayed. The GO button runs a robot simulation and the reset button loads a random design. The lower left panel is where the visualization of the robot movement is displayed.*

those in the control group could not see robots designed by others, while those in the experimental group could. Each user's objective was to create robots that were most amenable to behavior optimization by the evolutionary algorithm. The specific behavior we investigated was legged locomotion.

2.2.1 Robot Form

The robot's form was fixed but users could constrain different subsets of joints to move in phase with one another. This process in essence reduces the dimensionality of the search space for the evolutionary algorithm, as it only has to optimize movements for each joint subset, rather than all joints independently. We henceforth refer to each such user-generated set of joint constraints as a robot design.

The robot's form consisted of ten cylindrical segments, connected with nine hinge joints. The robot had four legs, each with both a "shoulder" joint and an "elbow" joint. Each of the four legs were connected to a "spine" (see Figure 2.2 for screenshots of robot). The hinge joints could sweep through a range of $[-45°, 45°]$. The "shoulder" joints rotated the legs through the transverse plane. The "elbow" and "spinal" joints rotated through the sagittal plane.

The robot did not contain any sensors: the evolutionary algorithm produced open-loop controllers for the robots. The robot joints were actuated using a displacement-control sinusoidal signal. This sinusoidal actuation resulted in several possible gaits, depending on the phase value of the sinusoid that drove each joint. The evolutionary algorithm described below could alter the relative phase between joints, but not the frequency or amplitude. The frequency was set such that the robot typically exhibited 40 rotations of each joint per evaluation. The amplitude allowed each joint to reach just about to its maximum and minimum range. However, this range could be constrained by the momentum of the robot and its friction with the ground plane.

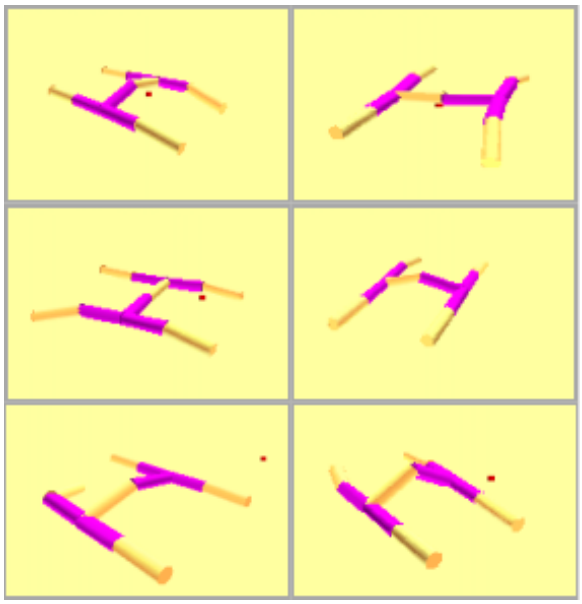

Figure 2.2: ***Screenshots the of robot.***

2.2.2 User Interaction

Using a graphical interface representing a top-view of the robot form, users drew lines that linked hinge joints at the robot legs and spine into a group. Each grouping of joints was assigned a distinct color, based on the user-drawn lines that visually linked the nodes representing leg or spine joints (e.g. see *Design Panel* in lower right corner of Figure 7.1). When users drew links between joints, they were forcing those joints to have the same phase offset of the sinusoidal control as all others in the joint group - i.e. the phase value in the sinusoidal displacement control signal was the same for all joints in a grouping. This visually-oriented, symbolic display of joint connections allowed for easy communication of intuition about movement of the robot. For example, a user might notice that connecting diagonal joints results in trotting. They

may realize that connecting the joints of the front legs together, and connecting the joints of the back legs together produces a two-beat version of cantering. The lines connecting joint group configurations acted as a symbol language that was designed to support the inherent pattern recognition abilities of the humans interacting with the underlying machine learning algorithm.

The robot movement was simulated with the *JBullet*[‡] physics simulation engine, a Java port of the *Bullet Physics Library*[§]. The user could design their joint grouping in an interactive drawing panel and then press the *Run* button to see a "heads up" or graphical display of robot movement simulated in the *simulation panel*.

The time allowed for each simulation was fixed at 20 seconds. Since there was some level of non-determinism in the simulation, we added *background runs* to increase the sample size for each of the grouping/phase configurations.

Each time a user clicked the "GO" button to launch a "heads up" run (an actual visual simulation of the robot movement), four other background runs with the same joint grouping design and the same associated phase offset values were run, which the user did not see. Fitness was set to the mean distance produced by these five runs. More background runs would have been preferable, but we were constrained by the need to maintain usability of the tool. Running multiple physics simulation processes at once required enough computing resources that more than four background runs might have degraded the user experience.

[‡]http://jbullet.advel.cz
[§]http://bulletphysics.org

2.2.3 Collaborative and Partitioned Groups

When a user opened the application, they were randomly assigned to either a control or experimental group. In the experimental group, users were allowed to see a subset of at most ten past designs created by other users in the experimental group. The ten designs were chosen randomly. If they returned to the site, they would see a different set of ten random designs. Users in the control group were limited to seeing only their own past designs in the history panel. If they ran more than ten designs, ten of their past designs were randomly shown. Users assigned to the control group would remain in the control group even if they left the site and then returned later, and similarly for the experimental group. Since users were anonymous, this group assignment was enforced using their IP address. As such, if a user were to use the platform from two different computers they could have been exposed to another group. The designs in the history panel were ranked from "worst" (left) to "best" (right) (by best average distance that was achieved for that particular design). The average distance moved by the robot with a given joint grouping was reported under that design. The number of times that particular design had been evaluated was reported above it. Users could use the designs in the history panel as inspiration for how best to improve on the designs and could contribute more runs to a particular design if they wished. They also had the option of generating a random design by clicking the Reset button and altering it into a new design of their making.

2.2.4 Evolutionary Algorithm

The initial values used for each joint group's phase-offset were randomly selected from a uniform distribution between zero and 2π radians. If a given design for a joint grouping was reused, a mutated version of the best past phase values was used in this new simulation. Mutation was implemented by picking a new value for each grouping with a probability of 0.1. Thus, the optimization used a hill-climber algorithm in which fitness (mean distance) was the basis for drawing from the population of eligible individuals (those that matched a given joint grouping). This best individual was then mutated. The distance that the robot was able to achieve with a given joint grouping and phase offset combination was interpreted as the fitness for that set of phase offsets. When multiple designs were created, the system became a parallel hill climber. Each design corresponds to an individual hill climber, and when a user creates a design that was previously created by another user (or re-created a design she has already created herself), another iteration of that hill climber was executed. If a user created a new design, a new hill climber with an initially random set of phase offsets was assigned to that design. The search space was thus a combination of both the combinatorial design space of joint group configurations and the real-valued space of phase-angle values to assign to each of those joint groupings.

2.2.5 User Incentivization

In order to incentivize users to participate in this experiment - and maintain their continued interest in using it - a sliding pointer under the *history panel* was incorporated

	Control	Experimental
Number of users	31	26
Number of runs	567	581
Number of designs	76	78

Table 2.1: User statistics

into the user interface. As the robot moved, its real-time distance was indicated by the sliding pointer relative to the visible historical designs in the *history panel*. This allowed the user to visualize how well their design performed relative to a subset of random past designs, either created by themselves (the control group) or their own and others' designs (the experimental group). A secondary function of the performance slider was to gamify [59] the platform. We hoped that the user would be motivated to either create new designs that outperformed past designs or provide additional computation to previous designs to improve on them. In short, the user was imbued with the sense that they were in direct competition with a subset of historical designs.

2.3 Results

A total of 57 anonymous volunteer users participated in the study via the web. The platform was advertised on online message board systems to attract participants. Participants were not financially rewarded for their participation. Of the users that volunteered, 31 were assigned to the control group and 26 were assigned to the experimental group. The control group ran the simulation a total of 529 times and the experimental group ran the simulation a total of 581 times (see Table 2.1).

A random sampling of designs created by users in both the control and experimental groups can be seen in Figure 2.5. We aggregated the maximum distance values each user was able to achieve. These maximum values were collected per group: the control (n=31) and experimental (n=26). The distributions of the control and experimental group were tested for rejection of the null hypothesis. The null hypothesis in this case was that the events come from the same distribution - that there is no difference between users who worked collectively versus those that worked individually. Using the t-test to reject the null hypothesis under the assumption of normality, we found significant cause to reject the null hypothesis at the 0.05 alpha level (p-value of 0.0404). Testing for the rejection of the null hypothesis without the assumption of normality also gave significant cause to reject the null hypothesis at the $\alpha = 0.05$ significance level, using the non-parametric Wilcoxon Rank Sum test (p-value of 0.0415).

Boxplots of the two distributions can be seen in Figure 2.3.

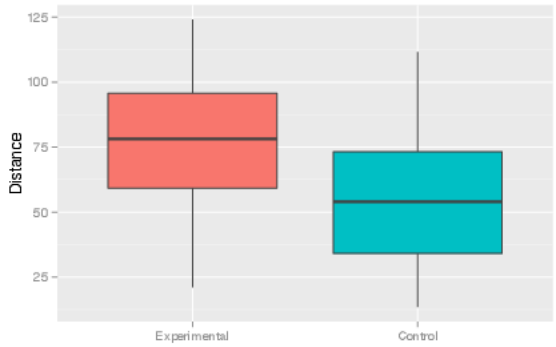

Figure 2.3: ***Distance distribution and p-values for significance between experimental and control groups.*** *Significance tests: t-test p-value=0.0406, Wilcoxon Rank Sum p-value=0.0418.*

2.4 Discussion

We have shown that the control and experimental groups are significantly different at the α level of 0.05. As such, we can say with some degree of confidence that the control and experimental groups are not drawn from the same distribution. That is, the maximum distances achieved by robots designed and optimized by those working collectively was significantly different than the maximum distances of robots designed and optimized by those working individually. Furthermore, the largest distances achieved by the group working collectively were higher than those achieved by the group of individuals working alone. The experiment thus shows that working in a cooperative setting improved the overall search for better robot designs under these experimental settings.

These results could be a consequence of the fact that the experimental group was exposed to more total designs as a whole. When users of the experimental group first opened the platform, they were exposed to designs from past users (with the exception of the first user). However, exposure to past designs by other users does not necessarily imply that the experimental group has an advantage. The hill climber assigned to a given design may have been able to perform more iterations in the experimental group compared to the hill climber assigned to the same design in the control group, because multiple users in the experimental group may have contributed computational effort to that hill climber. However, this would only advantage the experimental group if this extra effort was directed toward an eventually successful design. Groupthink may have led users in the experimental group to contribute ef-

fort to one initially promising design, yet caused them to neglect a design that was less successful at the outset but may have yielded a superior controller in the long run.

It is also possible that the designs to which the experimental group were exposed informed their design decisions, which led to improved designs. Isolating the cause for improved designs in the experimental group will be pursued in future work.

Our experiment involved two embedded search spaces: 1) the combinatorial assignment of joints to phase-locked groups, and 2) the assigning of phase offsets to each group. The users directly searched the first, combinatorial space and the optimization algorithm searched the second, real-valued space. The interaction between the two search spaces may contributed to a stronger separation between the control and experimental groups. Additionally, the simulation of robot movement was not entirely deterministic; although there was a clear separation of mean distances achieved by different configurations. A plot of distributions of one hundred runs for several random configurations can be seen in 2.4. There is a clear separation of distributions between "good" (large distance traveled) versus "bad" (small distance traveled) configurations as to inform the user's intuition. And this non-determinism and large search space was present in both the control and experimental population so each group operated close to the same conditions.

As is common in web-based usage, a small minority of designs attract the majority of computation [60]. We can see in Figures 2.6 and 2.7 that more users focused their attempts on those designs that were able to move farther. That is, more users ran

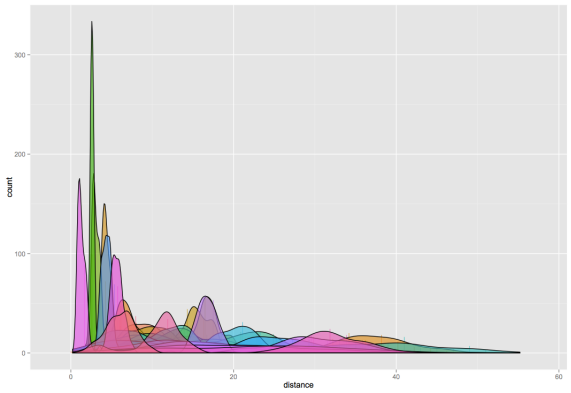

Figure 2.4: ***Distance distributions of robot movement for several joint group configurations.*** *Each configuration was run 100 times.*

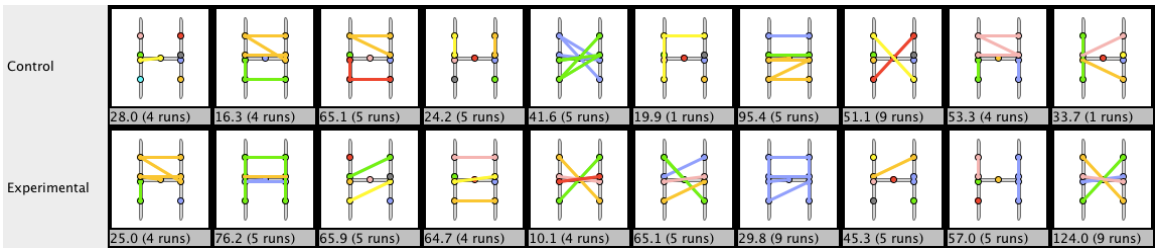

Figure 2.5: ***A random sample of designs created by users from the control and experimental groups, shown with best distance achieved and number of times a user evaluated that design (a "run").***

designs that were able to achieve higher distances in the experimental group; and more design evaluations were dedicated to higher fitness designs. Instead of focusing on designs that exhibited comical or low fitness behavior, users elected to add computing effort to promising, high fitness designs. Although there is not enough data to make definite conclusions about the possibility of "leap-frogging" (the tendency for users to draw inspiration from designs by other users to focus their efforts on promising designs) taking place in the experimental group, the data seem to suggest the possibility of positive feedback contributing to better performance when users worked collectively.

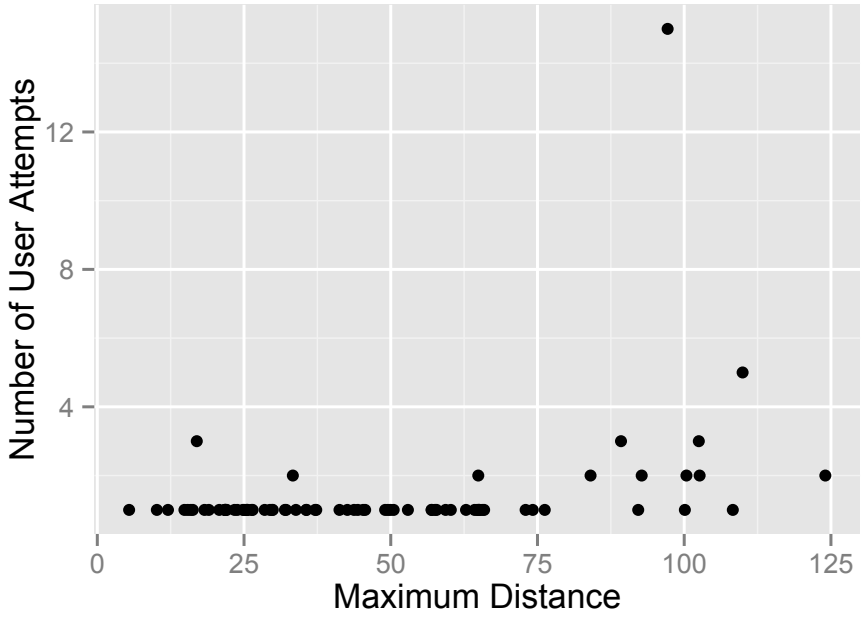

Figure 2.6: ***Number of users that attempted a unique design vs the best distance that that design achieved.*** *Each point corresponds to a design in the experimental group. Note that the design that received the most attention at 15 attempts was the "default" design that appeared when the application was launched in the user's web browser.*

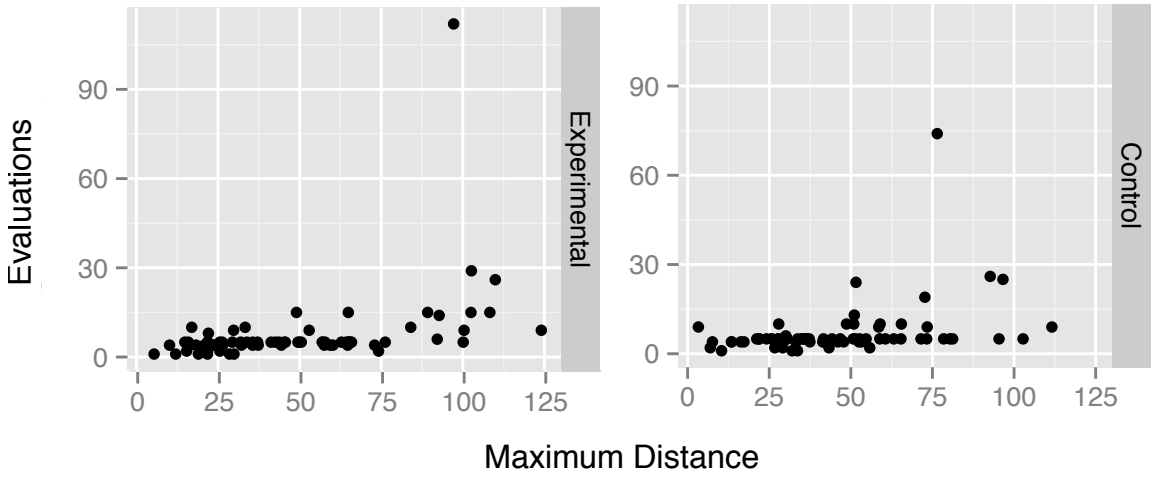

Figure 2.7: ***Number of evaluations for each design vs the maximum distance that design achieved.*** *As is the case in Figure 2.6, the design that received the most evaluations was the "default" design that appeared when the application was launched in the user's web browser.*

Using human subjects in evaluating fitness and developing new designs presents a number of challenges. When developing the platform, some effort was devoted to gamifying the system. The distance indicator was used to inspire the user to try and "beat" previous designs. Even though written instructions and the distance indicator were used to motivate the user to attempt to evaluate designs based on their ability to cause the virtual robot to move as far as possible, it is possible that users were either unmotivated by the preconceived goal of the simulation or had contrarian intentions to design robots that move as little or as comically as possible.

2.5 Conclusion and Future Work

We showed that in a hierarchical search space of robot designs in which human users searched the combinatorial level of the space and an optimization algorithm searched the real-valued level, a group that collectively searched the space was able to outperform a group of individuals that worked alone. The distributions of best robot distance achieved per person in each group were significantly different at an α level of 0.05 under both normal and non-parametric assumptions.

Using our framework, the user was shown the best distance that a given robot configuration moved as well as the number of runs contributed to each design. Novelty of a design might also be a useful message to communicate. Users might be incentivized to explore unknown regions of the search space if they were told that they came up with a design that no one else has tried. Additionally, the amount of symmetry in a given design or an indication of the spread of distances realized, such as standard deviation could be communicated. Social networks may be another means of incentivizing search: indicating which designs the user's friends have already explored could be built into the framework.

Another aspect of this work was the use of a symbolic language to easily communicate simple concepts of robot movement between human users. The language was so specific that it would only work with one form of robot. Future work on generalizing this language to quickly communicate complex ideas between collaborators may warrant further investigation.

Chapter 3

Combining computational and social effort for collaborative problem solving*

3.1 Abstract

Rather than replacing human labor, there is growing evidence that networked computers create opportunities for collaborations of people and algorithms to solve problems beyond either of them. In this study, we demonstrate the conditions under which such synergy can arise. We show that, for a design task, three elements are sufficient: humans apply intuitions to the problem, algorithms automatically determine and report back on the quality of designs, and humans observe and innovate on others' designs to focus creative and computational effort on good designs. This study suggests how

*Appeared as Wagy, Mark D., and Bongard, Josh C. "Combining Computational and Social Effort for Collaborative Problem Solving." PloS ONE. 10.11, 2015.

such collaborations should be composed for other domains, as well as how social and computational dynamics mutually influence one another during collaborative problem solving.

3.2 Introduction

Machine intelligence is arguably out-competing humans in a growing number of domains: autonomous robots are taking over warehouse [61], construction [62], and agricultural [63] tasks; machine learning methods are becoming adept at finding patterns of interest in massive data sets [64] as well as heretofore unknown biological relationships [65] and even physical laws [66] in raw data; and new search methods are increasingly challenging professional players in complex games [67].

As this process spreads and accelerates, it remains to be seen what role humans will play in an increasingly automated society. We present evidence that one key role that humans may continue to play is one of cooperation with machine intelligence: complementing the speed of machines with the human capabilities of creativity, pattern recognition, and an ability to apply intuitions about the physical world to abstract problems.

So far it has been shown that casual users can help in scientific domains such as protein folding [14], galaxy classification [15] and brain analysis [16]. Additionally, human participants in crowdsourced experiments have contributed pattern recognition capabilities to algorithms for aiding algorithms in generating realistic images [68] and defining effective robot control schemes [69,70]. However, the focus of these studies has been on the demonstration *that* it is possible to solve a complex problem by

casual participants working collaboratively on the Web rather than understanding *why* such a group of volunteers is successful at these tasks.

Our hypothesis is that humans can effectively work as part of a hybrid human-algorithm team by contributing their experience as embodied and social organisms to their algorithmic counterpart. To test this hypothesis, we created treatments that combined human participants and search algorithms together in different ways. Each team was responsible for designing and programming autonomous robots such that the robots performed a desired task, which in this study was rapid locomotion.

We chose robotics as the domain in which to study human-computer collaboration as it has traditionally been viewed as an extremely challenging enterprise: only small academic or industrial teams composed of individuals with advanced degrees have so far produced capable robots. However, work on crowdsourcing robotics [24,69,71] has shown that it is possible for casual users to accelerate the programming of autonomous robots and provide them with some semantic understanding of the world [72]. These studies make clear that humans can play an important part in human-computer interaction and human-robot interaction, but an understanding of why and in what ways this collaboration can be successful is still under-explored.

Moreover, we here report the first investigation into whether casual users can design robots, not just help them learn: In some of the teams we created, participants were tasked with designing robots that their computer could then program to move rapidly. We hypothesized that people may be able to bring their intuitions about animal movement to bear on this problem: casual users may, consciously or otherwise, know what kinds of body plans facilitate (or obstruct) the discovery of fast forward locomotion [73–75].

3.3 Methods

To understand how people and computers might work best together in this domain, we created three separate teams, each of which was tasked with designing and programming robots (note that we are here using the term "team" to distinguish between experimental treatments without necessarily referring to the standard social sciences definition of the term).

The first team was composed of human participants and algorithms: participants created, shared, and improved upon each other's robot designs (Fig. 3.1). [This work was exempted by the Committees on Human Subjects Serving the University of Vermont and Fletcher Allen Health Care, approval number 14-228.] Participants were recruited through the online bulletin board system, *Reddit*. We requested participation by querying users of several *subreddits*, pages devoted to subtopics of interest (such as artificial intelligence, robotics, programming and visualization). Therefore, participants were interested in technical subjects, but likely came from from varying backgrounds. However, users were anonymous so we cannot confirm the experience level of participants in the study with respect to each of these subjects. Reddit demographics are predominantly white, suburban males aged 18 to 29 years old [76]. All participants were unpaid volunteers.

The robots were virtual and behaved within a 3D simulated environment. Participants designed and observed their robots in a web browser (Fig. 3.1a) and were free to spend as much time at the task as they liked. They could also return and continue at any time. Each participant could also execute a search algorithm on their computer, which gradually improved controllers for their current robot (Fig. 3.1b,c). The

participant was free to dedicate as much or as little computational effort to a given robot as they liked, and could design as many robots as they liked. If the participant copied another participant's robot design, the new participant's computer continued searching from where the originating participant's computer left off (Fig. 3.1d). We will refer to this team as the *crowd/machine team* (CMT).

The second treatment was the same as the first, with one exception: participants could not see designs created by other participants. We will refer to this team as the *individual/machine team* (IMT).

3.3.1 Robot design by the crowd/machine and individual/machine teams

Participants who arrived at the experiment website were double-blindly placed at random into either the crowd/machine team (CMT) or individual/machine team (IMT) with equal probability. By comparing the performances of the CMT and IMT, we were able to address the question of whether participants spontaneously collaborate in this domain. Do they improve upon promising designs created by their peers, or do they become mired in group pathologies such as groupthink [38, 77]? Although recent work has begun to quantify conditions under which teams work well [78, 79], social collaboration within human/machine teams requires further study. For both teams, robot designs were simulated using a web-embedded physics simulation engine (`github.com/kripken/ammo.js/`). The physics engine is a Javascript-based open source version of the popular C++ physics simulation engine, Bullet (`http://bulletphysics.org/`). The simulation was rendered using

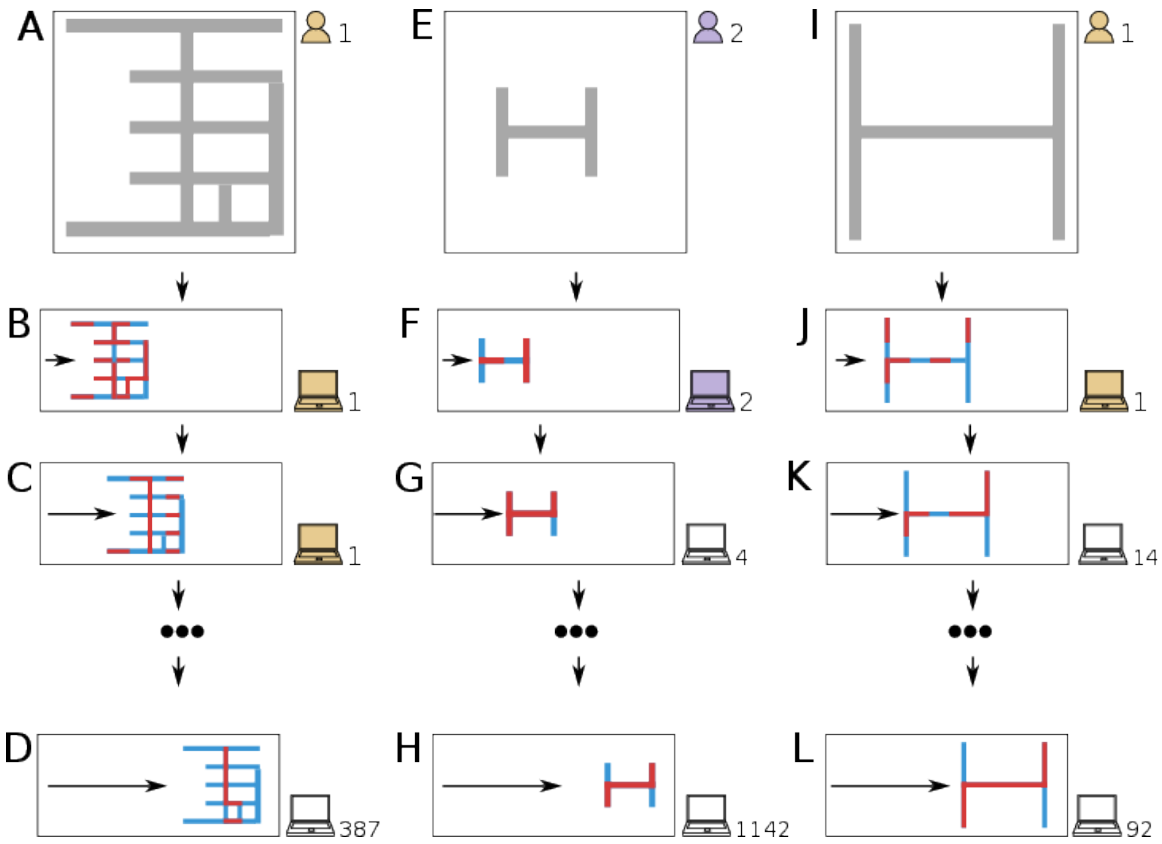

Figure 3.1: ***A hypothetical set of interactions in the crowd/machine team.*** *Participant 1 designs a robot (**a**) and then allows an optimization method to program his robot once on his computer (**b**). (The red and blue body segments oscillate in anti-phase with each other, resulting in a small amount of forward travel indicated by the gray arrow.) Participant 1 allows the optimizer to re-program the robot (**c**), hoping its behavior will improve. Participant 387 sees and likes this design, so she allows the optimizer to try again on her computer (**d**). Meanwhile, participant 2 designs a different robot (**e**) and performs one round of optimization on it (**f**). Later, several more participants also contribute computational effort to this design (**g,h**). After observing the quality of participant 2's design, participant 1 abandons his original design and attempts to improve on participant 2's robot by creating a larger variant of it (**i**). After performing an initial round of optimization on it (**j**), several more participants confirm the quality of this design (**k,l**).*

the WebGL graphics library (`www.khronos.org/webgl/`), a scene-based rendering library (`http://scenejs.org/`), and additional infrastructure code available online (`schteppe.github.io/ammo.js-demos/`).

Each participant was instructed to design a robot that could move as far as possible in the simulation. Participants accomplished this by designing a robot in the design panel (Fig. 3.2c), which was initially blank. They could then command a search algorithm to find good controllers for that robot. The quality of a controller is defined by how far it enables the robot to move from its starting position in fifteen seconds of simulation time. They could watch the progress of this optimization process in the simulation panel (Fig. 3.2b). Members of the IMT could see their own past designs in the history panel (Fig. 3.2a), while members of the CMT could see designs produced by themselves and other participants in the same panel. It was through this history panel that users 'communicated' designs to other participants.

Participants in either team were free to design as many or as few robots as they wished. They were also free to copy the designs produced by others (if they belonged to the CMT), create variants of other participants' designs, or create completely new designs.

Participants could connect a 5-by-5 grid of points with lines. They could draw as many lines as the grid allowed. If the participant connected two points neighboring one another horizontally, or two points neighboring one another vertically, they were connected by a single line. If the participant attempted to connect two points that were not vertical or horizontal neighbors with positions (x_1, y_1) and (x_2, y_2), those points were connected with a series of horizontal or vertical lines connecting pairs of horizontally or vertically-neighboring points. These lines were chosen such that they

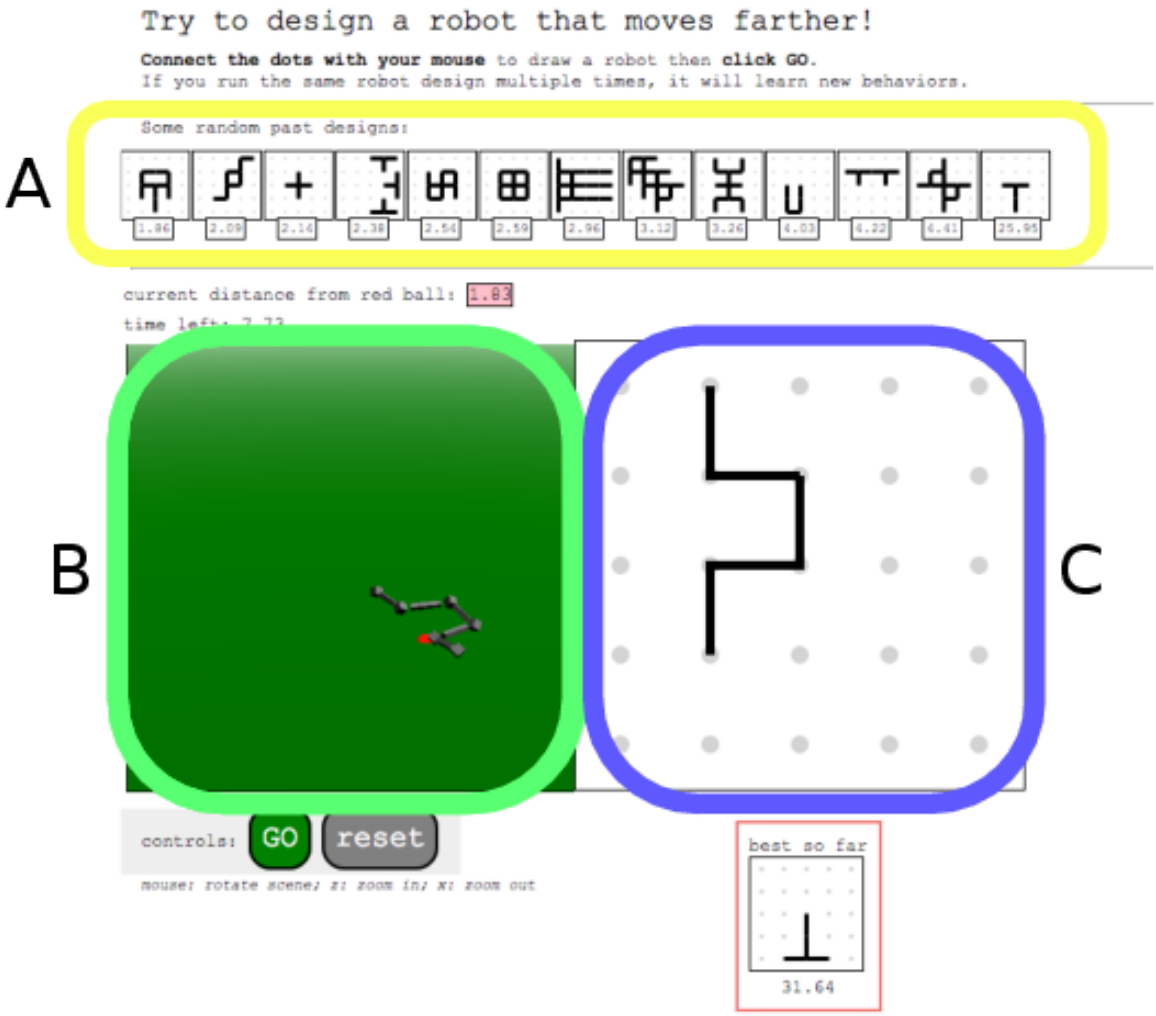

Figure 3.2: ***Screen-shot of the user interface with components.*** *Enclosed in the yellow box is the history panel (a); in green, the simulation panel (b); and in blue, the design panel (c).*

approximate as closely as possible the straight line connecting (x_1, y_1) and (x_2, y_2). This made diagonal and/or overlapping lines impossible, simplifying the process of converting these points and lines into a robot that can be easily simulated in the rigid-body physics engine. They could click the 'reset' button at any time to re-start

the design process.

Once the participant completed her design, the design would be rendered as a simulated robot using the physics engine. Each line of the design became a $1 \times 0.1 \times 0.1$ rectangular solid, representing one part of the robot's body. Each point adjacent to at least one line was instantiated as a $0.2 \times 0.2 \times 0.2$ cube in the physics engine. The cube was attached to each neighboring body part by a one degree of freedom rotational joint. The axis of rotation for each joint was set perpendicular to the plane defined by a vertical line passing through the center of the cube and the horizontal line passing through the center of the cube and the center of the body part to which it attaches. This axis of rotation was chosen to allow the robot to move outside of the plane defined by the grid. Segments were able to push the robot away from the ground plane to achieve locomotion, thus making the morphology an important factor in its ability to move rather than relying only on the friction of a sweeping motion along the ground plane using each of its segments.

The interface was designed to be both easy to use and intuitive, as participants were not expected to have any prior experience designing robots or using the tool. The interface enabled participants to design robots simply by "connecting the dots". Although this constrained design to a limited space of robots, the space was sufficiently large and sufficiently rich: there are $2^{(5 \cdot 4 + 5 \cdot 4)} = 2^{40} \approx 1.1 \times 10^{12}$ possible robots of differing sizes, symmetries, and biological realism, and which present varying levels of difficulty to the behavior optimization process. Each design produced by each participant of both teams was stored in a database on a central server.

3.3.2 Behavior optimization for the crowd/machine team and individual/machine team

Once a participant completed a robot design, she could click the 'go' button, which would dedicate some of her own computer's computational effort to finding a good controller for that design. When this button was pushed and the design did not yet exist in the central server's database, a hill climbing search algorithm was assigned to that design.

The hill climber improved behavior for its assigned robot design as follows. The hill climber searched over the space of possible controllers for that design, attempting to find those that enabled the robot to move as far from its starting position as possible. Every time a participant pressed the 'go' button, one iteration of the hill climber would be performed.

The first time the participant pressed 'go' for a unique design she had created, the number of one-degree-of-freedom rotational joints in the robot's design was counted. For a design with j such joints, a random binary string with j bits would be created. This string is assigned to the hill climber for this design. If a given design had received one or more iterations of search previously, then the current bit string associated with that design was copied, and each bit was flipped with 10% probability.

Then, the participant's design was rendered as a simulated robot in the physics engine as described in the previous section. The robot was allowed to move in the simulator for 15 seconds and $\sim$ 22 cycles of motor oscillations. At each time step, each of the j motors associated with each joint is controlled with position control. The desired position sent to each motor is determined by a sinusoidal signal. This

signal has a frequency of 1.5Hz and oscillates within $[-45^o, +45^o]$. All of the joints that have a zero associated with them in the current bit string oscillate in phase with one another, but are offset by a phase of π radians from those joints that have a one associated with them. The hill climber could in this way tune which joints move in phase or in antiphase with one another.

The participant could observe the resulting movement of the robot in the simulation panel (Fig. 3.2). If the participant clicked the 'go' button again, her computer would perform another iteration of the hill climber. The more times a participant clicked the 'go' button, the more search would be conducted for that robot design.

Each bit string for every design was stored on a centralized server. So, if a participant drew a robot body that had already been attempted by themselves or another member of her group, her computer would perform another iteration of search for that design, starting from the best controller found up to that point. This enabled members of the CMT to consciously collaborate on a common design. However, participants in both groups could also unknowingly contribute computational effort to an existing design if they were not aware that that design had already been created by another member of their group: only 13 historical designs were shown to a participant in the interactive robot design tool but many more designs were stored on the central server. If a user invoked a design that had already been drawn and run by another user in her own group (either the IMT or CMT), she would be continuing the hill climber for that design.

Two databases on the central server were established: one stored designs and controllers from the IMT, and the other stored designs and controllers from the CMT. This ensured that members of one team could not continue behavior optimization for

robots designed by members of the other group.

3.3.3 Communicating the results of design and optimization

The top of the interface housed the history panel, which displayed 13 pictorial and numerical summaries of past robot designs. Members of the IMT could only see their own past designs. Members of the CMT saw their own past designs, as well as those produced by other members of their team.

Each summary presented a top-down view of the robot's morphology, and the distance that that robot had managed to travel using one of the controllers that had been supplied to it by its hill climber.

Every time that a participant clicked the 'go' or 'reset' button, or refreshed the page, the 13 designs were erased. Each of the 13 slots was refilled as follows. For members of the IMT, one of their past designs was chosen at random. For the members of the CMT, any past design produced by members of that team was chosen at random. Then, one of the n controllers generated for that design by its hill climber was chosen at random from the central database. The robot design was drawn in the empty slot, and the distance traveled by that robot when controlled by that controller was written to that slot. The 13 summaries were then arranged in order, from the designs with the least displacement on the left to the most displacement on the right. This process was instituted to provide the participant with a sense for which morphologies may be amenable to controller optimization, and which morphologies may present the hill climber with a more multimodal search space.

Whenever a participant simulated a robot, its current displacement was reported in the simulation panel. If its current displacement matched or exceeded the distance achieved by a robot in the history panel, that number in the history panel would change from white to pink. This 'thermometer' metaphor gave the participant a real-time indication of how well their current design was able to move relative to the random sampling of past designs being displayed. It was hoped that this implicit competition might further incentivize participants to design robots with good potential for behavior optimization.

In addition to the thirteen randomly-sampled designs shown at the top of the interface, members of the IMT were shown their own best design at the bottom right of the page. The best design is defined as the one that had been most displaced by its hill climber, regardless of how many iterations of search each design has accumulated. Members of the CMT were shown the best design discovered by that team as a whole so far.

3.3.4 Robot design by the machine team

The third team we studied was composed only of computers: a state-of-the-art search algorithm [80] was employed to design robot body plans and to find good controllers for them. It has been demonstrated [81–85] that this algorithm outcompetes other algorithms at designing and programming robots. We will refer to this team as the *machine team* (MT).

We selected the Compositional Pattern Producing Networks–Neuroevolution of Augmenting Topologies (CPPN-NEAT) algorithm [86] for this, as it has been shown to be superior to other search algorithms for finding efficient gaits for robots [84] and has

also been employed for optimizing both robot morphology and control [81–83, 85, 87]. Another reason we chose this algorithm is because it was originally designed to mimic biological development's bias towards the production of regular patterns. This bias toward regularity has been implicated in CPPN-NEAT's ability to outperform other algorithms in automatically designing high-performance complex artifacts [86].

The CPPN-NEAT algorithm extends a simple evolutionary simulation, which at any point in time manages a population of Compositional Pattern Producing Networks (CPPNs). Each CPPN is capable of generating a regularly-patterned design within a user-defined, finite n-dimensional space. The user must also supply an objective function that can be used to score the quality of the resulting design. When the algorithm runs, CPPNs that produce designs with low objective function scores are deleted from the population, while CPPNs that produce high-performing designs are copied, randomly modified, and placed into the recently-emptied slots in the population. The algorithm continues for a user-specified number of generations. Finally, there are a number of experimental parameters that must be set before the algorithm can be run. For the work reported here we employed the CPPN-NEAT implementation available at `www.multineat.com`. We employed the same parameter settings as those used in [84].

Each CPPN generates a robot design as follows. Each CPPN takes four real-valued inputs and produces two binary outputs. From the five by five grid of points in the experimental website, each pair of horizontally neighboring points is selected in turn. For each pair, the horizontal and vertical coordinates of that point pair are input to the CPPN. The first binary output value is then read out. If the value is zero, the next point pair is supplied to the CPPN's inputs. If the value is one, a line

is constructed between those two points, just like a human participant might draw a line connecting that point pair. After each pair of 20 horizontally-neighboring points is supplied to the CPPN, the positions of each pair of 20 vertically-neighboring points is supplied to it. This resulted in the creation of zero or more horizontal and vertical lines connecting some of these point pairs together.

This process enabled each CPPN to design one robot. It was possible that some CPPNs create 'null' robots comprised of zero lines. Such robots were automatically assigned an objective function value of zero.

3.3.5 Controller generation by the machine team

The CPPN created a controller for its robot as follows. For each CPPN, each pair of horizontally-neighboring and vertically-neighboring points connected by a line is iterated over. If each of these point pairs have positions (x_i, y_i) and (x_j, y_j), the CPPN is supplied with the position where one of the two resulting one-degree of freedom joints would be (x_i, y_i) and the position of the midpoint of the line connecting the point pair $(\frac{x_i+x_j}{2}, \frac{y_i+y_j}{2})$. The bit at the CPPN's second output is then read out: a value of zero indicated that that motor would rotate the body segment centered at $(\frac{x_i+x_j}{2}, \frac{y_i+y_j}{2})$ about the cube centered at (x_i, y_i) with a phase phase offset of zero radians, while a value of one indicated that that motor would rotate with a phase offset of π radians. For the same point pair, the CPPN's inputs were now supplied with positions (x_j, y_j) and $(\frac{x_i+x_j}{2}, \frac{y_i+y_j}{2})$. The bit arriving at the CPPN's second output now dictated the phase with which the motor at (x_j, y_j) would rotate the body segment at $(\frac{x_i+x_j}{2}, \frac{y_i+y_j}{2})$ relative to the cube at (x_j, y_j).

The processes described in this and the previous sections thus enable a CPPN to

generate both the morphology and control of a single robot.

3.3.6 Behavior optimization for the machine team

When the CPPN-NEAT algorithm is initiated, it generates a population of P randomly-generated CPPNs. Each CPPN in the population is evaluated in turn. The CPPN produces its robot, the robot is evaluated in the physics engine, and the robot's resulting distance from its starting position is assigned as that CPPN's objective function score. Higher-scoring CPPNs are copied, mutated, and replace lower-scoring CPPNs in the population as described in [86]. The new entrants in this next generation of CPPNs are evaluated, and this process continues for a set number of generations. The number of generations was chosen so as not to exceed the number of evaluations that was reached by the CMT.

Because different variants of the CPPN-NEAT algorithm were conducted with differing population sizes, the number of generations G for each variant was set to the floor value of the number of evaluations performed by the CMT as a whole (11998) divided by the population size: $\lfloor \frac{11998}{P} \rfloor = G$. We performed 100 independent replicates of this algorithm and measured the average displacement of the robots produced by the best CPPNs in each population, at each generation.

In order to ensure the robustness of our results, we investigated the performance of the CPPN-NEAT algorithm using five sets of experimental parameters as reported in Table 3.1.

Fig. 3.3 reports the performance of the MT using the parameter settings in Case D. This set was reported because it achieved the highest mean performance of the five settings.

Case	Population (P)	Mutation Rate	Representation	Generations (G)	Replicates
A	100	0.05	4 inputs	119	100
B	100	0.05	2 inputs	119	100
C	100	0.1	4 inputs	119	100
D	200	0.05	4 inputs	59	100
E	50	0.05	4 inputs	239	100

Table 3.1: Machine team experimental settings investigated.

In addition to varying the experimental parameter settings for the CPPN-NEAT algorithm, we also investigated two different representations for the generation of robot morphology. We considered both a four-input representation described above as well as a two-input representation.

In the two-input representation, each CPPN has two instead of four inputs. To create a robot, the position of the midpoint between each of the 20 vertically-neighboring and 20 horizontally-neighboring grid points was input to the CPPN in turn. The output of the CPPN was interpreted in the same way as the four-input CPPN representation. The robot controller was created by inputting the coordinate of the midpoint between the center of the cube to which a given body segment was attached and the midpoint of that segment into the CPPN to determine whether the corresponding motor command had a phase offset of zero radians (an output of zero) or a phase offset of π radians (an output of one).

3.4 Results

We found that robots designed by both human teams on average outperformed the designs created by the MT (Fig. 3.3). Moreover, despite the fact that more com-

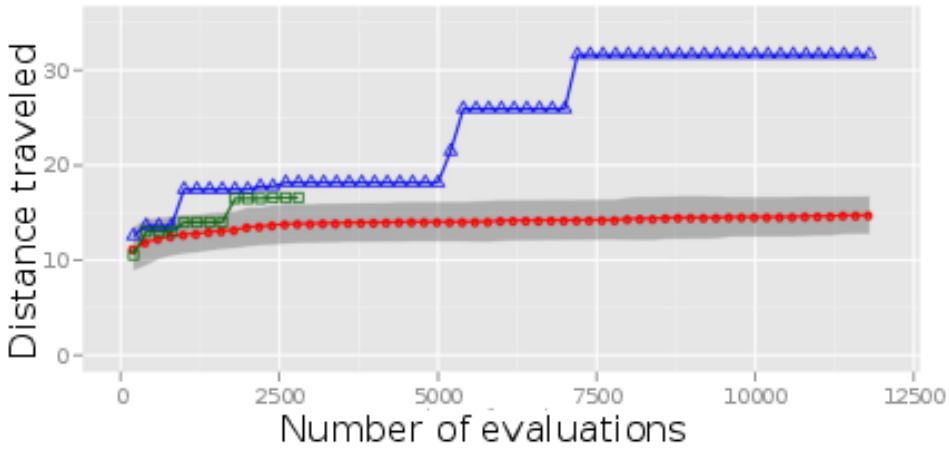

Figure 3.3: ***Comparing how the three treatments designed robots.*** *The relative design abilities of the MT (red), IMT (green), and CMT (blue). The unit of distance u is set equal to the length of one robot body segment. The markers for the IMT and CMT report the furthest displacement achieved by any robot produced by that team up until that point. The markers for the MT report the mean displacement achieved by the 100 best robots, one drawn from each of the 100 independent trials of the search algorithm employed by this team. The gray band reports the 95% confidence interval for the MT.*

putational effort was expended by the MT than by the CMT, no design generated by the MT was able to outperform the best design created by the CMT. Using the complementary cumulative probability distribution of MT performance at the final evaluation, we found that there was near zero probability that the MT would produce a better design than the CMT (probability less than 0.0001). The greatest distance achieved by a robot out of all runs and all treatments of the MT was 28.0 units as compared to the maximum distance of 31.7 achieved by the CMT.

Participation numbers for both the CMT and the IMT are summarized in Tables 3.2 and 3.3.

The top performing robots for each group are shown in Fig. 4.5.

The individual/machine team portion of the experiment was stopped when we saw an appreciable decrease in participation (number of participants on day 1: 2805,

	IMT	CMT
Total evaluations	2839	2839
Total participants	367	384
Total designs	1246	1090
Average controllers per design	1.64 ± 2.17	1.67 ± 2.94
Designs per participant	4.56 ± 5.96	4.54 ± 4.99
Performance of the top 30 designs*	14.49 ± 2.79	16.48 ± 4.10
Perfectly symmetric designs (among the top 30 designs)	27	22

*Table 3.2: Comparison of the aggregate behavior between two human/machine teams. *Participants who collaborated produced superior designs compared to those who worked individually (Welch's t-test; p=0.036, DOF=49.4, t=-2.16; random normally distributed independent samples)*

	MT	CMT
Total evaluations	11800	11800
Total participants	—	947
Total designs	1897 ± 118	2885
Average controllers per design	1.83 ± 0.08	2.40 ± 4.98
Designs per participant	—	5.63 ± 7.13
Performance of top designs	14.75 ± 1.01	31.64
Perfectly symmetric designs (among the top 30 designs)*	1	25

*Table 3.3: Comparison of the aggregate behavior between one of the human/machine teams and the machine team. *The CMT produced significantly more perfectly symmetric designs than the MT.*

day 2: 2108, day 3: 615, day 4: 164, day 5: 117, day 6: 3). The crowd/machine team was allowed to run longer for the sake of comparison to the machine team, giving the MT an opportunity to catch up to the CMT performance.

3.5 Discussion

What caused the superior performance of the human/machine teams over the MT? The MT finds a robot that is able to achieve a distance of 28.0 units, which is near the

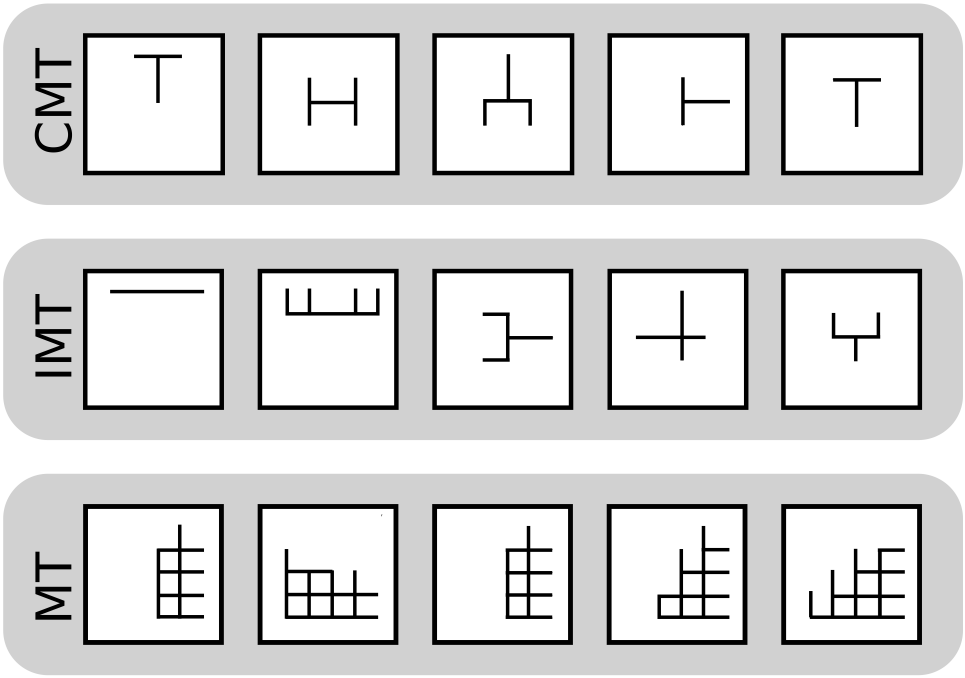

Figure 3.4: ***Top*** 5 ***robots in each of the treatments: the crowd/machine team (CMT), the individual/machine team (IMT) and the machine team (MT).***

best found by the CMT of 31.7 units, but this is the best result of 100 independent runs each of 5 different treatments explored with the MT: a much larger number of attempts than the CMT was allowed. Although the CMT evaluated more designs than the MT, we do not expect the superior performance to be due to these differences. This is because, in comparison, the IMT designed more robots than the CMT, yet performed worse than the CMT (Table 3.2). The MT did use CPPN-NEAT to define both the morphology as well as the control algorithm for the robots in contrast the the hillclimber that was used to define the control of the robots in the CMT and in the IMT. However, we hypothesize that using CPPN-NEAT for control versus defining the control scheme with a hill climber is in fact an advantage that the MT has over the human/machine teams based on past work [84].

In addition to creating what appear to be simpler designs (smaller number of segments and connections between them) as seen, for example, in Fig. 4, a prominent feature of the designs created by the human/machine teams is the presence of symmetry (Table 3.3; see Supporting Information S1 Text for discussion of symmetry measure). Even though the search algorithm employed by the MT is known to be advantageous to other search methods of its type because it biases search towards more symmetric patterns [80], we observed a significantly higher number of designs that were perfectly symmetric among the top designs produced by the CMT compared to the top designs found by the MT. Moreover, human/machine teams focused on symmetric designs much more than would be expected, as there exist only two perfectly symmetric designs per million in the space of 1.1×10^{12} robot designs made possible through the user interface. Crowd participants focused on symmetric bodies at the outset of the experiment as well as the termination. This serves as evidence that the participants came to the experiment with a bias toward symmetric designs, even before being exposed to the performance of their own or others' robot designs. The presence of perfectly symmetric designs created by participants was just as prevalent in the first 30 IMT designs as in the last IMT 30 designs: a significant number of these first designs were perfectly symmetric (23 of the 30, $p = 0.0067$; 1-sample proportions test, χ-squared=7.50, df=1; samples independently distributed); nearly the same number in the last 30 designs created by participants were perfectly symmetric (22 of the 30, $p = 0.01762$; 1-sample proportions test, χ-squared=5.63, df=1, samples independently distributed).

Bilaterally symmetric robot designs may be advantageous because, if coupled with a symmetric controller, they are better able to produce directed locomotion than

bilaterally asymmetric robot body-plans, regardless of controller [88]. To test this we constructed a physical version of the best robot produced by the CMT, and ran it using the best controller found for that robot by the CMT, which also happened to be symmetric (Fig. 3.5a-i). We compared the performance of this physical robot against two variants: the same robot with randomly-generated asymmetric controllers (Fig. 3.5j-l), and an asymmetric robot composed of the same number of parts and controlled by randomly-generated asymmetric controllers (Fig. 3.5m-o).

We found that the symmetric controllers significantly outperformed the asymmetric controllers (Mann-Whitney U test with Bonferroni correction for multiple comparisons; $[p = 0.0045, p = 0.0275, p = 0.0316]$, $[W = 88, W = 79, W = 78.5]$, independent distributed samples) and, in all cases, the symmetric robot and controller outperformed the asymmetric robot with asymmetric controllers (Mann-Whitney U test with Bonferroni correction for multiple comparisons; $[p = 0.0002, p = 0.0002, p = 0.0003]$, $[W = 100, W = 100, W = 98.5]$, independent random samples). This suggests that symmetric robot body plans may be advantageous for this task, and that part of the explanation for why the CMT outperformed the MT is because its members were cognizant of this fact at some level of awareness, or became aware of it to some degree by viewing the quality of the designs produced by other members of their team.

In addition to a preference for symmetric designs, we investigated whether collaboration influenced the performance of the CMT. To do so, we compared the top designs produced by the CMT with those produced by the IMT.

This investigation was complicated however by the fact that varying numbers of robot body/controller combinations were evaluated by each of the CMT and IMT. In some cases, only a single controller was evaluated for a robot design, whereas other

robot designs attracted many controller evaluations by members of a team. Thus, to better estimate the quality of robot designs produced by the IMT and CMT, we investigated in more depth how amenable to behavior optimization their best designs were. To do so, we first drew the top 50 ranking designs from both IMT and CMT according to the best controller found for those designs within their originating teams. We then performed 100 replicates of a genetic algorithm [89] against each design. Each genetic algorithm replicate was given a population size of 100 with a 10% mutation rate and 10% crossover rate. Genomes were bit-strings that indicated either a zero-phase (0) or π-phase (1) controller and the fitness objective was identical to that used in the web-based tool – a simulated robot was to move as far as possible within the allotted fifteen seconds of simulation time. The maximum displacement achieved by the design for each replicate was extracted and averaged over the 100 replicates. The resulting relative performance of the teams' top k designs for various values of k is reported in Table 3.4.

k	**IMT distance moved**	**CMT distance moved**
20	14.4 ± 3.0	17.0 ± 4.1
30	14.5 ± 2.8	16.5 ± 4.1
40	14.7 ± 2.9	16.7 ± 4.1
50	14.6 ± 2.8	16.7 ± 4.1

Table 3.4: Distance moved by the top k robots after intensively optimizing their controllers. (Welch's t-test; $p < 0.05$ for all values of k.)

Despite only slight differences between the aggregate behavior of the different teams (Table 3.2), the CMT's best designs outperformed the IMT's best designs (Welch's t-test; $p < 0.05$ for all values of k [$k = 20, p = 0.029, DOF = 33.3, t = -2.29; k = 30, p = 0.036, DOF = 49.4, t = -2.16; k = 40, p = 0.038, DOF = 51.3, t = -2.13; k = 50, p = 0.031, DOF = 50.8, t = -2.22$], samples independent

and normally distributed). The only difference between the two human/machine teams was that members of the CMT could see pictorial representations of their own and others' designs, and the distances achieved so far by those robot designs, whereas members of the IMT could only see their own designs, and the distances those designs had so far achieved. This indicates that the superior performance of the social over the IMT must be due to collaborative innovation: members of the CMT avoided poor designs, dedicated more computation to promising designs, and/or created variants of good designs more than members of the IMT. In fact, the collaboration on designs can be seen in the average and maximum number of unique user contributions to designs in Table 3.5. Although groupthink and other group pathologies [40] may have been present in the CMT, their corrosive effects, if present, were outweighed by the positive feedback mechanisms of collaboration through non-verbal communication. This finding suggests that such collaboration may be harnessed in other design domains if appropriate pictorial representations, coupled with simple reports of the quality of each design, can be formulated.

	CMT	**IMT**
Maximum Number of Users per Design	210 *(out of* 919*)*	20 *(out of* 331*)*
Maximum Proportion of Users per Design	0.229	0.060
Mean Number of Users per Design	1.798	1.278

Table 3.5: Unique user contributions to designs. The crowd/machine team resulted in more users contributing to individual designs than in the individual/machine team, suggesting that the crowd did contribute collectively to some designs.

3.6 Conclusion

Here we have shown that a team composed of human designers who can dedicate more or less computational effort to their own or other participants' robot designs outperformed an machine team lacking human members as well as a treatment composed of human members who could not collaborate. Although we are not the first to demonstrate that human and algorithm teams can outcompete machine-only groups [14–16], we are the first to present reasons for why this occurs for design tasks. We showed that individuals bring pre-existing intuitions to bear on one part of the problem, while machines determine, through search, whether these intuitions are born out.

This intuition favored designs with symmetry, a morphological attribute that results in fast forward locomotion. That the crowd/machine team outperformed the individual/machine team demonstrates that social processes have a beneficial effect in this domain. While we cannot confirm which social processes resulted in the improved outcome, we hypothesize that this beneficial social collaboration derives from one or a combination of the following factors.

First, our interface allowed participants to view the work of others. We hypothesize that allowing participants to strengthen their pre-existing intuitions by viewing the work of others may have benefitted the crowd/machine team. Based on these growing intuitions, the participants could dedicate more or less computational effort to vet designs using search algorithms. The search algorithms in turn expose incorrect assumptions or validate intuitions. Second, human-produced designs and their machine-generated quality estimates were advertised to the group using a representation interpretable by casual users. If our interface did not represent the designs in

this way, the improved performance of the crowd/machine team over the individual/-machine team would have been possible only by chance, which we demonstrated to be unlikely (see Discussion section). Whether the participants benefitted from seeing the human-generated designs, the machine-generated quality estimates, or both is unknown, but at least one of the two must have been responsible for the superior performance of the crowd/machine team over the individual/machine team.

Thus, by capturing intuitions from casual users and exploiting synergistic social processes arising among them, we have shown that casual users may contribute useful creative work to a collaboration between humans and machines.

Given more time and computational resources, we would be interested in extending the analyses to account for other parameterizations of the evolutionary algorithms used in the study. Due to resource limits, we needed to define a somewhat arbitrary cutoff on the duration of the experiment and work with a fixed set of parameters in evolutionary algorithms. Had we investigated other parameters, we might have found results that were different from the findings of the present study. Additionally, the optimization method used may have converged upon a locally optimal result, thus missing the global optimum. It could be that the set of globally optimal robot designs are those that are not symmetric and the human participants exposed the algorithm to a locally, but not globally, optimal bias.

Future studies will focus on the social dynamics of the crowd interactions and the means by which human designers were influenced by both their interaction with other members of the crowd and their own experimentation with the design tool. In this study, we observed that it was beneficial for the human-algorithm variant to be exposed to other human designers. However, the modes that this social dynamic

are beneficial may or may not be similar to the social dynamics present in other studies. Users may have, as the result of their own experimentation or by observing the design of other participants, discovered morphological variations that work well in tandem with a particular control structure; or they may have been able to improve substantially on a design by another participant that they might not have discovered on their own.

Additionally, vetting users according to their expertise would be an interesting addition the analyses. While we suspect that the majority of users in the present study were non-experts, the participants were anonymous and thus we cannot verify the extent to which the users were experts or non-experts.

Our finding suggests that increasingly large, diverse and complex collaborations that combine people and machines together in the right way (and further empowered by collaboration-enabling web tools, cost-effective additive manufacturing [90] and cultural trends such as the Maker movement [91]) may accelerate innovation in a wide range of fields. Finally, such work may help ensure that accelerating technological advancement develops into an empowering rather than a disenfranchising phenomenon.

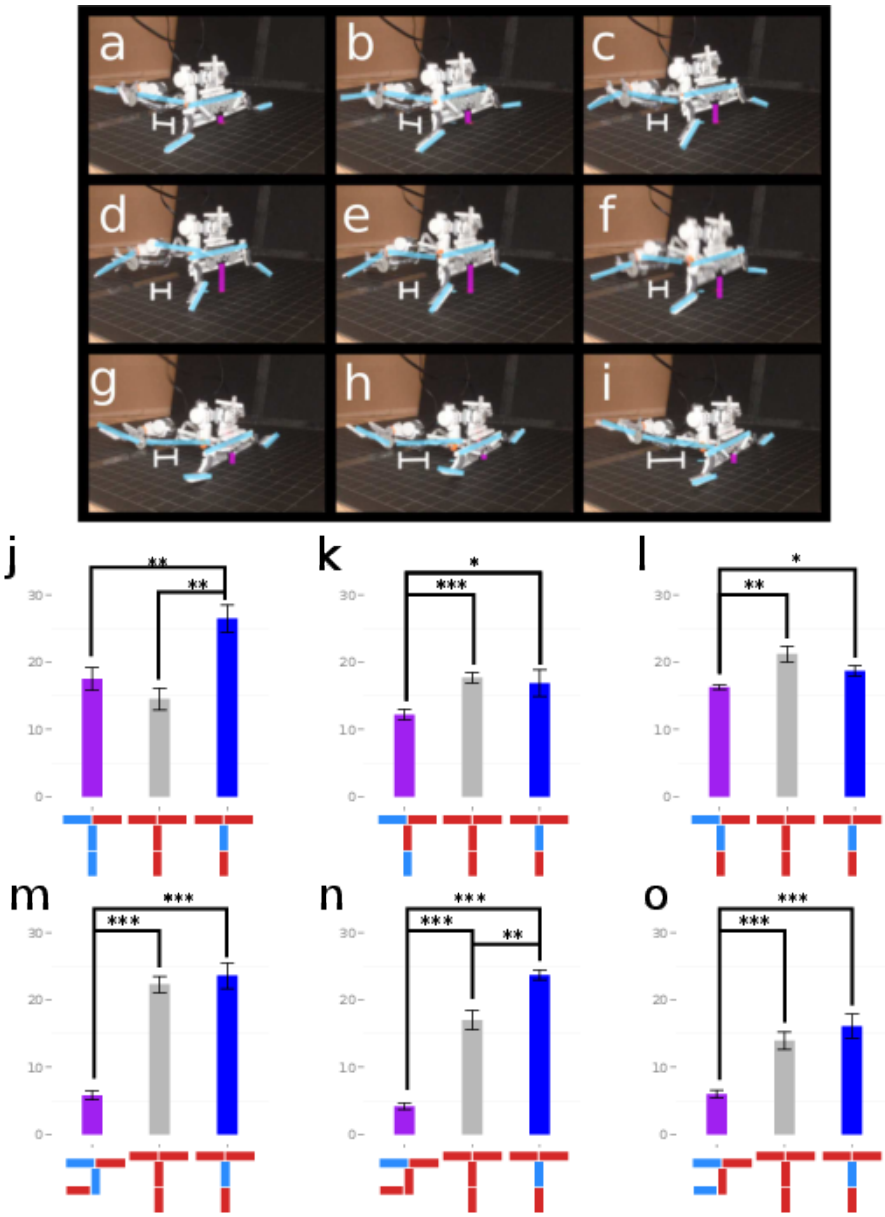

Figure 3.5: ***Performance of the physically constructed robot developed through the collaborative effort of 209 participants. a-i****: gait of the physical realization of the best robot found by the CMT, and Panels* ***j-l*** *report the average distance travelled by the physical robot when equipped with the controller generated by the CMT (blue), three different randomly-generated asymmetric controllers (purple), and a randomly-generated symmetric controller (gray). Each bar in these panels reports average distance achieved over 10 independent trials. Panels* ***m-o*** *compare the average distance travelled by a randomly-generated, asymmetric, physical robot with the same topology as the symmetric robot (purple) to a randomly-generated symmetric controller on the symmetric robot body generated by the CMT (gray) and the symmetric controller discovered by the CMT (blue).*

Chapter 4

Social Contribution in the Design of Artifacts on the Web*

Abstract

The Web has created new opportunities for interactive problem solving and design by large groups. In the context of robotics, we have shown recently that a crowd of non-experts are capable of designing adaptive machines over the Web. However, determining the degree to which collective contribution plays a part in these tasks requires further investigation. We hypothesize that there exist subtle yet measurable social dynamics that occur during the collaborative design of robots on the Web. To test this, we enabled a crowd to rapidly design and train simulated, web-embedded robots. We compared the robots designed by a socially-interacting group of individuals to another group whose members were isolated from one another. We found that there exists a latent quality in the robots designed by the social group that was significantly less prevalent in the robots designed by individuals working alone. Thus, there must exist synergies in the former group that facilitate this design task. We also show that this latent quantity correlates with the desired design outcome, which was

*To appear as Wagy, Mark D. and Bongard, Josh C. "Social Contribution in the Design of Adaptive Machines on the Web". ALIFE 16: The Sixteenth Conference on the Synthesis and Simulation of Living Systems. Vol. 16, 2016

fast forward locomotion. However, the quantity – when distilled into its component parts – is not more prevalent in one group than another. This finding demonstrates that there are indeed traces left behind in the machines designed by the crowd that betray the social dynamics that gave rise to them. Demonstrating the existence of such quantities and the methodology for extracting them presents opportunities for crafting interfaces to magnify these synergies and thus improve collective design of robots over the web in particular, and crowd design activities in general.

4.1 Introduction

The Web has led to novel modes of social participation [92] and means for contribution to tasks that were previously limited to small groups of experts [14, 15, 33]. New Web technologies, such as WebGL 3D graphics and Web-embedded physics engines, have made the interactive design of intelligent machines possible over the Web [93]. Additionally, collective intelligence methods [6] such as crowdsourcing [7], human computation [9] and social computing [13,94] can be used to incorporate contributions of large groups of non-experts – the 'crowd' – into the design of robots on the Web [69]. However, the ways that people involved in design and problem-solving tasks synergize remains an open question.

Under certain circumstances, collectives have computational abilities not readily available to the group members in isolation [3]. However, the ability of a group of people to socially coordinate problem solving efforts has limits in physical social interactions [95]. These limitations have also been shown to persist in Web interactions [96].

We seek to better understand whether a crowd of humans interacting socially through the Web can contribute non-destructively, and potentially in a superadditive way [37] to collective problem solving. Whether, and in what ways, constructive social computing phenomena arise in human interactions on the Web remains to be seen. Previous studies [14, 15, 33] have demonstrated that a crowd of individuals can complete problem-solving and design tasks while working in parallel. However, understanding the ways the crowd can collectively exhibit abilities that differ from that of an aggregate of individual contributions is under-explored. The present study

addresses how the contributions of the crowd as a social entity can be measured as distinct from the contributions of isolated individuals working in parallel.

We have shown that the crowd is capable of collectively designing adaptive machines on the Web [69, 97]. Thus there is evidence that under some circumstances social synergy can arise in this domain. However, past studies have not uncovered the imprint left behind on the crowd-generated designs that result from this synergy.

We hypothesize that this social contribution is a measurable quantity. If this quantity is indeed measurable, then it could be actively managed. If it constructively contributes to the task at hand, it can be enhanced. If it is destructive, it could be actively suppressed. In this study, we seek to demonstrate whether or not the quantity is measurable; and if so, whether it is constructive or destructive with respect to the design of robots. Developing automated means for the discovery of crowd contributions is the first step to actively manage crowd participation towards productive, collective outcomes.

Evidence indicates that humans may be biased, during design, by exposure to the physical environment in which we are embodied. For example, people may favor symmetric robot designs as symmetry is ubiquitous in nature. If incorporated into the physical characteristics of artificial organisms, these biases facilitate design [97, 98]. However, we still do not know whether these biases arise individually or if the bias is reinforced by crowd behavior. If these biases are the result of social pressures, they could be actively enhanced or suppressed depending on whether they were beneficial to the desired design outcome.

Previous work suggests there may exist as-yet undiscovered latent contributions from the crowd [99]. However, the methods suffer from a deficiency: they obscure

underlying features of the crowd-generated designs that may contribute to a positive outcome. Additionally, we do not know whether design features were indeed influenced by social processes. This stands in contrast to the present work, in which we demonstrate a method for deriving contributions as linear combinations of simple geometric values.

In the present work, we distill a latent geometric factor from robot designs using Singular Value Decomposition (SVD). We demonstrate that this value is more prevalent in a social design process than it is in designs resulting from the parallel effort of isolated individuals. Furthermore, we show that this latent factor has a beneficial effect on the objective of the design process.

4.2 Methods

In this study, we used a dataset of user-contributed robot designs [97]. Participants designed robots using an interactive tool available through their web-browser. When a user visited the study website, they were shown a 5×5 grid of dots. By clicking on a dot and dragging to another dot, they could draw line segments between dots (e.g., Fig. 4.1). Line segments were only allowed between horizontally- or vertically-adjacent dots to constrain users from crossing segments in their designs. In order to enforce this constraint, the closest set of adjacent lines segments that approximate the diagonal line drawn by the user was shown as the user dragged lines between dots. For example, a line drawn from the top left dot to the bottom right dot was approximated by a zig-zag formation of smaller, adjacent segments between the top left and bottom right dots.

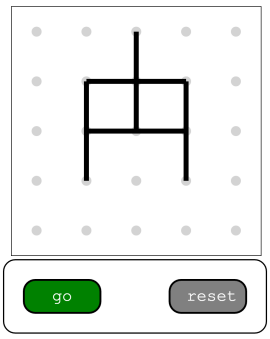

Figure 4.1: ***Grid of dots for designing robot bodies by the crowd. An example design is shown drawn on the grid.*** *Users could click on a dot and drag to another dot. Lines were only allowed between adjacent dots. If a user dragged to a dot that was not adjacent, the path of adjacent lines that best approximated the dragged path was generated as the designer drew the line.*

When a participant was finished designing a robot, she could click a "GO" button, which launched an instance of her design as a 3D robot in a physics simulation engine within her browser. The simulation contained only the robot and an infinite, flat ground upon which the robot could walk.

Line segments drawn in the grid of dots were translated into the physics simulation as 3D rectangular parallelepiped robot segments. Each dot that was adjacent to at least one line segment was invoked as a 3D cube in the physics simulation. Segments adjacent to a cube were connected to the respective cube with a hinge joint along the axis at the midpoint between adjoining cube and segment faces and in parallel with the ground and these body faces. In this way, the robot was able to push in the direction of the ground. However, as segments flexed and the robot body moved, the robot configuration did allow for sweeping motions across the ground by its members.

Every joint was actuated with a sinusoidal, displacement-controlled signal. All sinusoidal signals driving the hinge joints swept the same angle ($\pm 45°$) at the same frequency (1.5 Hz). However, the phase of the signal could take on one of two possible

values: 0° or 180°. A hill-climber search algorithm was used to define which of these phase values was assigned to each of the hinges in the robot created by the user. We will use the term *phase configuration* to refer to a single assignment of phase values to hinge joints for a particular robot body. A separate hill-climber algorithm was maintained for each robot design and for each group. Thus the first instance of a particular robot morphology within a group was assigned a random phase configuration. When that same user or another participant in the same group drew and simulated an identical robot morphology, the robot was assigned a slightly altered version of the original phase configuration. Thus, each time a user clicked *GO*, they contributed one iteration of the hill-climber algorithm for a particular robot body.

Each robot design was simulated for 15 seconds of physics simulation time (for examples of robots in the web-embedded physics simulation, see Fig. 4.2). The distance that the robot traveled in these 15 seconds was recorded in a database along with the adjacency matrix that defined the connections.

Participants were asked to design a robot that could move as far as possible across the flat plane and within the allotted 15 seconds of simulation time. However, participants were unpaid volunteers so they were free to use the tool however they desired.

Participants were placed into either a control group or experimental group with an equal probability of being placed into either group. Participant IP addresses were recorded so that if they were to return to the site to design more robots, they would be placed in the same control or experimental group. In a panel at the top of the study website, the experimental group was shown 13 randomly chosen (2D) designs created by other users in their group. We refer to this group of robot designers as

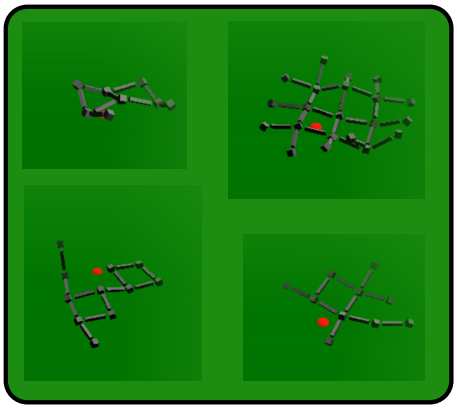

Figure 4.2: ***Robots were simulated in a web-embedded physics simulator for 15 seconds.*** *Each line that a user drew was translated into a segment in the robot and each dot that was adjacent to a line was translated into the physics simulation as a cube. The cubes and segments were attached with a hinge joint, which was actuated by a motor with displacement-controlled sinusoidal signal.*

the *social group* because they were given the chance to utilize other users' designs if they so desired. Every time a user returned to the site, they would be shown a potentially new random selection of 13 designs created by other users that were placed in the experimental group. The control group, which we will refer to as the *independent group*, was shown only their own past designs. Thus the user interfaces for the control and experimental groups were identical apart from the content of the panels showing historical designs at the top of the site.

When the crowdsourced portion of the experiment was complete, we used the collected robot designs to compare the design preferences for users in the experimental and control groups.

Since robot designs consisted of a series of points and edges joining these points, we were able to compute network metrics on the resulting dataset and derive a set of simple geometric measures from each robot designed by the crowd. These geomet-

ric measures included minimum, average and maximum degree measures; maximal matching; length of the shortest path; node connectivity; number of legs; number of segments; radius; transitivity; number of cliques; indicators of bipartiteness, regularity, whether the network is a tree and biconnectedness; and symmetry (computed as the maximum proportion of segments that are matched with another segment across either the horizontal, vertical or one of two diagonal axes of symmetry in the 2D design plane).

We then distilled this set of computed geometric measures for each robot morphology into just one representative value of its geometric features. We did this by using the Singular Value Decomposition (SVD) [100] dimensionality reduction technique to factorize the matrix of all descriptive geometric features into component singular values and matrices. We then used only the first singular value of the decomposition to obtain a one-dimensional representation of the geometric robot feature-space, which allowed us to represent each design by a single number that reflected all computed geometric features into one quantity. This reduced-dimensional representation of robot morphologies will henceforth be referred to as the latent factor representation of a particular robot's morphology, or the *latent factor* in short.

4.3 Results

A sample of designs created by participants can be seen in Fig. 4.3. The T-shaped robot design in the center was the highest performing design overall (the best distance it was able to achieve was approximately 32 meters).

Summaries of team contribution are indicated in Table 4.1.

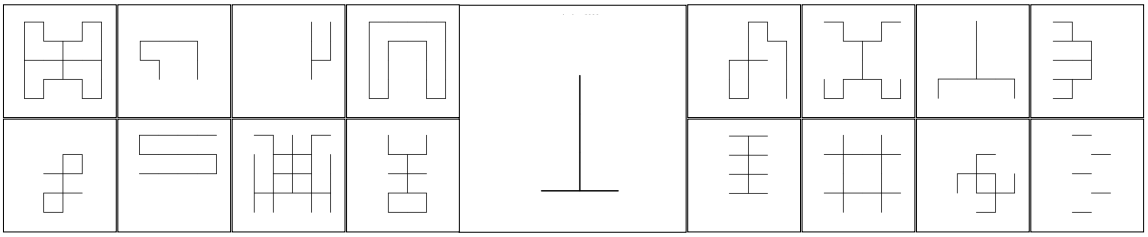

Figure 4.3: ***A sample of robot designs by participants.*** *The highest performing robot (with regard to distance-traveled) was the T-shaped robot in the center of the designs shown.*

	Independent	**Social**
Total Contributions	2825	2984
Total Unique Designs	1245	1136
Number of Participants	364	398

Table 4.1: Social and independent group contributions.

The distribution of distances achieved by the social group and that of the independent group can be seen in Fig. 4.4. The social group achieved significantly higher distances than the independent group ($p < 0.0001$; Kolmogorov-Smirnoff test).

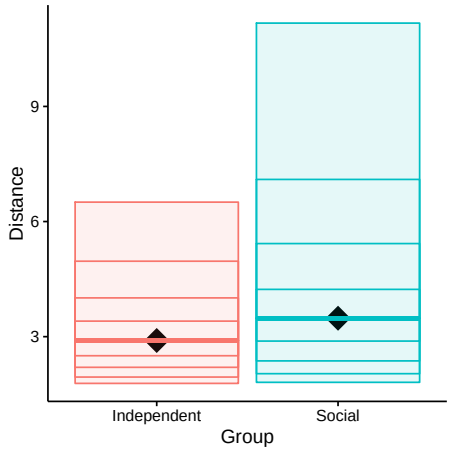

Figure 4.4: ***Deciles of distances achieved by robot designs in the group working together (Social) and those achieved by individuals working in isolation (Independent).*** *The median distance value is indicated by a black diamond.*

The minimum, mean and maximum values of the latent factor and the quantities

that are used to compute it are shown in Table 4.2.

	Independent (min/mean/max)	**Social** (min/mean/max)
Number of Limbs	(0/0.734/12)	(0/0.656/12)
Symmetry	(0/0.882/1.00)	(0/0.945/1.00)
Latent Factor	(0/0.880/1.10)	(0/0.942/1.11)

Table 4.2: Latent factor range and ranges of values used to compute latent factor in designs by both groups.

The dimensionality reduction technique resulted in two non-negligible components (coefficients > 0.00001) that contributed to the latent factor. These contributions were the robot's *number of legs* (coefficient of 0.01) and the *symmetry* of the robot body (coefficient of 0.99). Values for symmetry could range from a minimum of no symmetry (0.0) to a maximum value of 1.0, indicating perfect symmetry about at least one of the horizontal, vertical or diagonal reflective axes. The minimum number of legs found in a design was 0, representing designs whose segments were all connected at both ends to up a maximum value of 12 legs.

There was not a significant difference in the symmetry of the designs created by the independent group compared to those created by the social group ($p \approx 0.132$; Kolmogorov-Smirnoff test), nor was there a significant difference between the distribution of the number of legs in designs created by the social group and the independent group ($p \approx 1.0$; Kolmogorov-Smirnoff test). However, there was a significant difference in the values for latent factor when comparing the distribution of designs created by the independent group and the social group ($p < 0.01$; Kolmogorov-Smirnoff test).

The 5 designs with the highest latent factor value can be seen in Fig. 4.5.

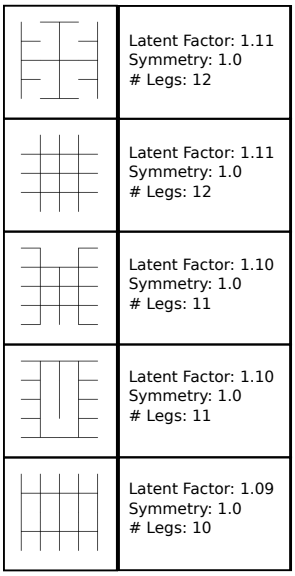

Figure 4.5: ***Crowd designs with the highest latent factor values and corresponding symmetry and number of legs calculations.***

4.4 Discussion

The only difference between the social group and the individual group was that the social group was able to see designs created by others in their group. Thus, if a quantity derived from the designs was more prevalent in the social group versus the independent group, then this quantity was the result of social dynamics.

We derived a measurable latent factor from designs created by each group. We found that this latent value was significantly more prevalent in the social group than that in the individual group at a statistically significant level. Thus exposure to others' designs resulted in increased prevalence of this factor.

A synergy is, by definition, a collective outcome that is greater than the sum of individual contributions. Demonstration of a synergy is not predicated on the collective outcome being constructive with respect to the objective of the efforts.

However, there is evidence that the latent factor that we have distilled contributes constructively to robot design.

This latent, socially-transmitted value incorporated morphological symmetry, which has been shown to be beneficial for robot locomotion [101]. While the incorporation of symmetry in user designs alone was not significantly different in the social versus independent groups, the associated p-value ($p = 0.132$) indicates a trend that those in the social group may have favored symmetry in designs over those working independently. It is only by the incorporation of the number of legs in the design that differentiates the social tendency of design with the isolated design tendency.

It appears that there is a latent quantity that – through exposure to designs by other users – is increasing, consciously or otherwise in the social participants. And the social group does indeed design robots that, on average, outperform those designs by participants in the individual group. However, we cannot say with confidence that it is this latent value that leads to the improved performance. There may be other, undiscovered factors that lead to the superior performance by the social group. However, we did find that there is a positive correlation between the discovered latent factor and the distance that a robot is able to travel (Pearson correlation ≈ 0.255). Thus, while we cannot say for certain that we found *the* latent factor that contributed to the success of the social group, we can say that – through the distillation of design decisions by a large group of non-expert contributors – we found a quantity that correlates with the desired problem outcome. And that this quantity is, in part, corroborated by findings scientific literature [101] on locomotion unbeknownst to the non-expert participants in the study.

The social group designed robots that were capable of moving significantly farther

than those designed by individuals in isolation. As can be seen in Fig. 4.4, the distribution of distances achieved the social group's robots have a higher median value than the independent group. The distribution of distances achieved by the social group has a much higher spread in the upper deciles than the independent group. Here we are reporting the overall ability of the participants in the social group to the ability of the independent group to design robot bodies in tandem with their control strategy. This is in contrast to the performance of the robot morphologies that are independent of their control strategies as described in [97]. Thus users were able to work together to build robot body/brain combinations that outperformed the body/brain combinations of those that worked alone. This need not have been the case: well-known group pathologies such as groupthink [38] could have resulted in an echo-chamber effect. Users working together could have missed promising design possibilities due to a focus on limited regions of the space of possible designs. Also, the number of participants working collectively in this study (398) far exceeds the number of participants considered optimal for effective social interactions [95].

However, there could be a trivial reason for the social group's superior design performance. As described in the Methods section, a hill-climber was assigned to each robot morphology; and each hill-climber instance was shared between members in a group. It is possible that the designs that received the most attention by the social group simply were given more attempts at finding a good control strategy by devoting more hill-climber iterations to the search for a good controller. Thus groupthink may have led the social group to focus on a single design's control strategy rather than concentrating efforts on finding an optimal morphology. If this had been the case, we would have seen that the designs with the most hill-climber iterations are also the

highest performing designs. However, this is not what we see in the results. Referring to Fig. 4.6, we do see that several designs in the social group benefited from many more hill-climber iterations than the independent group. But these designs were not among the high-performing instances. In contrast, many of the highest-performing designs are those that received 10 or less hill-climber iterations. This is much less than some designs, which received in excess of 100 iterations devoted to finding the best controller. Thus we can see that the robot morphology – not just the number of hill-climber iterations – was an important component of the robot's performance. Therefore we believe that the social group did not rely solely on the focused search efforts for the best control strategy on a limited groups of designs.

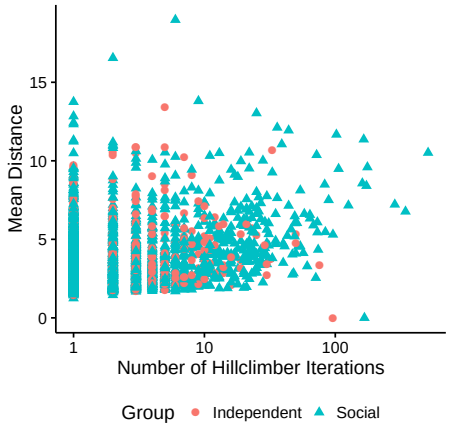

Figure 4.6: ***Mean distance achieved by each robot design compared to the number of hill-climber iterations that that design received.*** *A number of designs in the social group did receive substantially more iterations than those in the independent group, but they were not high-performing designs. In contrast, some of the highest-performing designs received very few iterations.*

We claim here that the latent factor discovered is an indicator of synergy in the social group. This value is significantly more prevalent in the social group versus independent group. And the only difference between the two groups was the opportunity

to synergize. Therefore the only ways that the latent factor could arise is through synergy or by chance. It is unlikely that the value arose by chance, as indicated by the statistical tests performed. However, it has been shown that crowdsourced work follows a heavy-tailed distribution [102]. Most participants contribute very little to a crowdsourced activity and a single user or small group of users contributes vastly more than others. This study followed that same trend (see Figure 4.7). Therefore, it is possible that by chance the top contributing user in the social group favored the latent factor and biased the overall group towards prevalence of this value.

We investigated whether the designs by the top-contributing participant in the social group biased the comparison of the latent factor between the groups. We removed the social group member that contributed the most designs from the social group and compared the mean latent factor values for users in each group. Even without this top contributor, we found significantly more of the latent factor in the social group versus the independent group ($p = 0.011$ Kolmogorov-Smirnoff test).

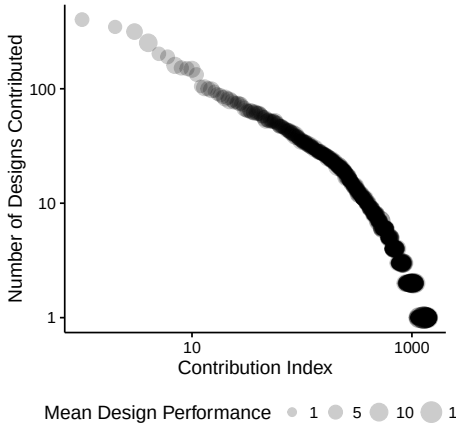

Figure 4.7: ***Contributions to this study follow a heavy-tail distribution.*** *The number of designs contributed by most users was small whereas the number of designs contributed by one very enthusiastic user was very high.*

Note that the designs with the highest scores of the latent factor are those that maximize both the symmetry measure – a maximum of 1.0 – and the number of legs. The maximum number of legs found in designs created by the crowd was 12. 12 legs is also the maximum number of legs possible when designing single-component, connected robots in this design space. In fact, 8 unique members of the social group drew designs that maximized the number of legs quantity; whereas no users in the individual group drew designs that maximized this quantity. The maximum number of legs found in the independent group was 11. Thus, the designs that maximize the latent factor (Fig. 4.5) are those with the most legs possible. However, despite having the highest possible symmetry score, the top performing design of all designs only has 3 legs (the largest, T-shaped design in Fig. 4.3). We cannot say that the best design was the result of this increased latent factor in the social group. Whether the prominence of the latent factor influenced the creation of the best designs requires further exploration. Future work will address the progression of features that lead to specific, optimal outcomes such as the T-shaped robot in Fig. 4.3.

We can get a sense of the participation of users that contributed to the top designs by looking at Table 4.4. Repeating numbers across rows in the table indicate that the same user contributed multiple hill-climber runs to a particular design. Two patterns can be seen in these top 10 designs. The top design (Design Rank = 1) ranking design was created by a user that only contributed 9 runs total to the overall experiment. Similarly, the first five contributions of the sixth and eighth ranking designs were by the same user. However, in almost all other designs in these top 10 designs (with just one exception: the seventh ranked design), we see that the designs were created by a user with lower numbers of total contributions and then picked up by users who

contributed more overall to the experiment. For example, the second best design was created by a user who contributed just 4 runs to the experiment. Then a user that contributed 6 runs picked up the design and then a user that devoted 25 runs drew that same robot to contribute a run to the hill-climber. This pattern of contributions could be the result of various social dynamics. It may be that a design that is initially promising may have caught the attention of those users that are more participatory. Or this could be a general pattern of behavior for any design created socially. However, if we examine the same table for the worst-performing designs (Table 4.3 the pattern is not as prevalent. But we do see this pattern in the eighth and ninth worst designs. Future work will investigate these social dynamics that contribute to specific robot designs.

Rank	**User 1**	**User 2**	**User 3**	**User 4**	**User 5**
1	9	9	9	9	9
2	4	6	25	25	25
3	3	17	17	88	88
4	6	29	52	52	52
5	52	52	52	52	316
6	28	28	28	28	28
7	17	14	14	14	14
8	11	11	11	11	11
9	8	8	8	83	79
10	10	13	75	75	75

Table 4.3: Total number of runs contributed by each of the first 5 *users to work on the top ranking designs. Repeating numbers indicate the same user contributing runs to the design. The best design is at the top and the tenth best design is at the bottom.*

Additionally, the mean value of this latent factor over the course of time (as measured in number of designs contributed by each group) can be seen in Fig. 4.8. This figure strongly suggests a slow increase in the social group's usage of the latent factor, whereas it appears that this value fluctuates over time for the independent

Rank	User 1	User 2	User 3	User 4	User 5
10	133	133	133	133	133
9	38	148	148	148	148
8	23	79	79	79	79
7	7	4	6	6	15
6	35	35	35	35	35
5	50	50	50	50	50
4	23	23	23	10	252
3	52	52	52	52	52
2	46	46	46	46	46
1	41	41	41	41	41

Table 4.4: Total number of runs contributed by each of the first 5 users to work on the bottom ranking designs. The worst design is at the bottom and the tenth worst-performing design is at the top.

group. More data is required to evaluate whether this trend continues in order to verify that this is indeed a trend; but the steady increase of this quantity in the social group is suggestive.

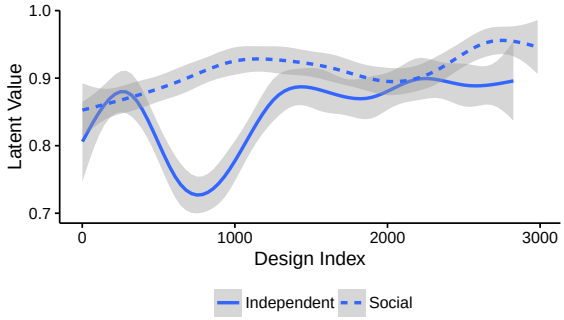

Figure 4.8: ***Mean value of latent factor over time (index over designs created by participants).*** *Error bars indicate 95% confidence levels. Participants creating designs socially maintained a near constant increase in the latent value, whereas participants working independently varied their incorporation of this value in their designs.*

4.5 Conclusion and Future Work

In this study, anonymous Web participants designed robots in their browsers. Our hypothesis was that there exists a measurable quantity derived from the social design of adaptive machines. Using the designs created by study participants, we derived a latent geometric quantity through a dimensionality reduction technique. We compared the prevalence of this derived latent value in the social and individual groups. We found that this value was more prominent in a group of participants working socially than those working in isolation. We observed that this value followed an increasing trend in the social group designs whereas the quantity fluctuated in the designs created by the independent group.

The latent value that we uncovered is derived from the symmetry and number of limbs in robot designs. That this value was derived in part from symmetry corroborates previous work on social design of adaptive machines over the Web. The latent value was shown to have a positive correlation with the desired outcome in robot designs, which was that of fast forward locomotion. Thus, the social quantity may have played a role in the superior outcome in social design of robots. Further work is required to demonstrate how such aggregate social quantities influence specific designs, such as those that are among top performers.

Deriving such measurable social quantities can be useful for their incorporation into automated methods. In future work, we will use crowdseeding [99] to enable machine design of robots by incorporating the SVD-derived objective into a a design objective.

Additionally, we would like to utilize the methodology introduced here to analyze

crowd designs in order to provide immediate feedback to the crowd. By providing feedback to the crowd during the social design process, we may be able to capitalize on crowd preferences and biases in real-time to thus accelerate the search process.

This technique may be useful more broadly in social and human computing. But understanding whether social design preferences are destructive or constructive in other domains is critical to determining the general utility of such methods.

Chapter 5

Crowdsourcing Features in a Predictive Model*

Abstract

Crowdsourcing has been successfully applied in many domains including astronomy, cryptography and biology. In order to test its potential for useful application in a Smart Grid context, this paper investigates the extent to which a crowd can contribute predictive hypotheses to a model of residential electric energy consumption. In this experiment, the crowd generates hypotheses about factors that make one home different from another in terms of monthly energy usage. To implement this concept, we deployed a web-based survey within which 627 residential electricity posed 632 questions that they thought would be predictive of energy usage. This same group then provided 110,573 answers to questions posed by their peers. Thus users both suggested the hypotheses that drove a predictive model and provided the data upon which the model was built. We used the resulting question and answer data to build a predictive model of monthly electric energy consumption, using random forest regression. Because of the sparse nature of the answer data, careful statistical work was needed to ensure that these models are valid. The results indicate that the crowd can

*In review as Wagy, Mark D., Bongard, Josh C., Bagrow, James P., Hines, Paul D.H. "Crowdsourcing Predictors of Residential Electric Energy Usage". IEEE Transactions on the Smart Grid.

generate useful hypotheses, despite the sparse nature of the dataset.

5.1 Introduction

With the rapid adoption of Advanced Metering Infrastructure (AMI), end-users have an increased ability to track their electricity consumption [103–105].

However, making the transition from merely tracking energy usage to making better decisions about how and when people use electricity (responsive demand) is likely to require additional feedback. Responsive demand requires that end-use consumers and energy managers make informed choices about which investments and behavior changes will have the most substantial impact on their energy bills. Prior work [106] has shown that consumers often misunderstand the relative impact of various loads on electricity usage and that information feedback can produce substantial reductions in energy usage [107].

There is thus a need for tools that leverage smart meter data to help consumers understand their electricity usage. One approach to this problem is for utilities to use expert-driven models of energy consumption [108] and then use these models to provide feedback to consumers. However, many customers distrust information from utilities, which can substantially reduce the effectiveness of utility-driven demand-side management programs [109]. An alternative to expert-driven processes is to provide end-users themselves (i.e., the crowd) with tools that enable them to discover useful patterns through a collaborative process. Only very preliminary work has investigated the potential of crowdsourcing for helping residential customers to understand electricity usage [110, 111]. It is not known whether consumers, who are not experts in energy efficiency modeling, can add value to expert knowledge of how behavior drives residential energy usage.

In this study, we employ a crowdsourcing technique described in [112,113] in which crowd participants ask questions that they believe drive some behavioral outcome. Specifically, in this study we ask users to contribute questions that they believe are predictive of electric energy usage. In addition to asking questions, these same participants are given the opportunity to answer questions contributed by their peers. From these questions and answers we build predictive models that indicate which behaviors are relevant in modeling energy usage. Based on these models, we ask: can a crowd of non-experts contribute to the process of hypothesis formulation about energy usage behaviors?

5.2 Methods

In the present study, we used the methodology introduced in [112] to identify predictors of behavioral outcomes. This method proceeds as follows: First, participants are recruited to visit a website focused on understanding a behavioral outcome. Next, the participants are asked to answer a few questions. These are questions that others had previously suggested, which they believe to be effective predictors of the outcome of interest. For example, one might believe that obesity is related to eating habits and thus ask, "How many meals do you eat per day?" In the background a modeling engine works to identify relationships between the resulting answer data and the outcome of interest, and then communicates this information back to the participant.

In this paper we will describe the results from an application of this method to the problem of providing information feedback to residential electricity consumers. Specifically, our "EnergyMinder" application was designed to use AMI data from a small

municipal utility, the Burlington Electric Department (BED), and the crowdsourcing method above to answer the following question: "Why does one home consume more electric energy than another?"

This experiment began by designing a web site, which was made available to about 15,000 customers in Burlington, VT in the fall of 2013. After initially logging in to the site, customers could view their electricity consumption compared to that of others in the participant group, and then were invited to both answer and pose questions regarding residential power usage in the manner previously described. Once a user posed a new question, that question was forwarded to the moderators who verified that the question was not egregiously misleading and did not include personally identifying information, and then added to the crowdsourced survey. This process was seeded with a set of six expert-generated seed questions (see Table 5.1). Participants were free to answer and ask as many questions as they desired. Upon arriving at the "ask" page, the site prompted participants with the suggestion that they ask questions that they believed to be predictive of residential electricity usage. Users were not limited to answering or asking questions in a single session; they could return to the site as many times as they wanted to answer or ask questions.

The site allowed participants to pose three different types of questions: questions with numerical answers (e.g., "How many loads of laundry do you do per week?"), yes/no questions ("Do you have access to natural gas?"), and agree/disagree questions ("I generally use air conditioners on hot summer days."), which were based on a five-level Likert scale with the option to *Strongly Disagree*, *Disagree*, *Neither agree nor disagree*, *Agree* or *Strongly Agree*. To initiate the question/answer process, a set of six seed questions, shown in Table 5.1, were inserted into the EnergyMinder tool.

Table 5.1: Expert-generated seed questions

ID	Text
q1	I generally use air conditioners on hot summer days
q2	Do you have a hot tub?
q3	How many teenagers are in your home?
q4	How many loads of laundry do you do per week?
q5	Do you have an electric hot water tank?
q6	Most of my appliances (laundry machines, refrigerator, etc.) are high efficiency.

In addition, the site included a "virtual energy audit" page that was designed to provide some feedback to customers about factors that appeared to cause their energy consumption to deviate from the mean. To implement this, the site used a forward-only step-wise AIC (Akaike Information Criterion) linear regression approach to build a model of the form:

$$\Delta E_{30,i} = \sum_{j \in A} \beta_j z_{j,1} + \epsilon_i \tag{5.1}$$

where $\Delta E_{30,i}$ is the deviation of customer i's energy consumption from the mean, A is the set of questions selected by the AIC algorithm, β_j is the estimated coefficient for question j, $z_{i,j}$ is the mean (0) imputed z-score of user i's answer to question j after dropping outliers such that $|z_{i,j}| < 3$, $\forall i, j$, and ϵ_i is the unexplained energy. $\Delta E_{30,i}$ was formed by summing the past four weeks of energy consumption data for user i, collected anonymously from the utility's AMI system, multiplying by 30/28 to obtain a rolling one-month outcome value and then subtracting from the mean for the participant group.

To reduce the risk of over-fitting, the model was constrained to include no more

than 20 terms. Given the model (5.1), the energy audit page for participant j displayed at most 10 questions for which the terms $|\beta_j z_{i,j}|$ were largest for this customer. This page also included an illustration of how much their answers to these questions impacted their predicted energy usage. In this way, users could see how their usage differed from usage patterns of an average consumer using questions and answers that had been found through the crowdsourcing process.

5.2.1 Differences from standard survey research

This method of crowdsourcing survey information differs from standard survey-based research in that the participants are both generating the survey questions as well as answering them. When a participant poses a new question, they are essentially proposing a new crowd-generated hypothesis regarding behaviors that they believe may affect residential energy consumption by asking questions and provides data for predictive models by answering these questions. Thus a collaborative process exists in the way that the crowd participates: the number of questions are ever-growing as are the answers in response to that growing body of questions (see Fig. 5.1). In order to differentiate this type of crowdsourcing from the more common technique in which participants are asked to accomplish a fixed set of tasks, we call this process 'collaborative crowdsourcing'.

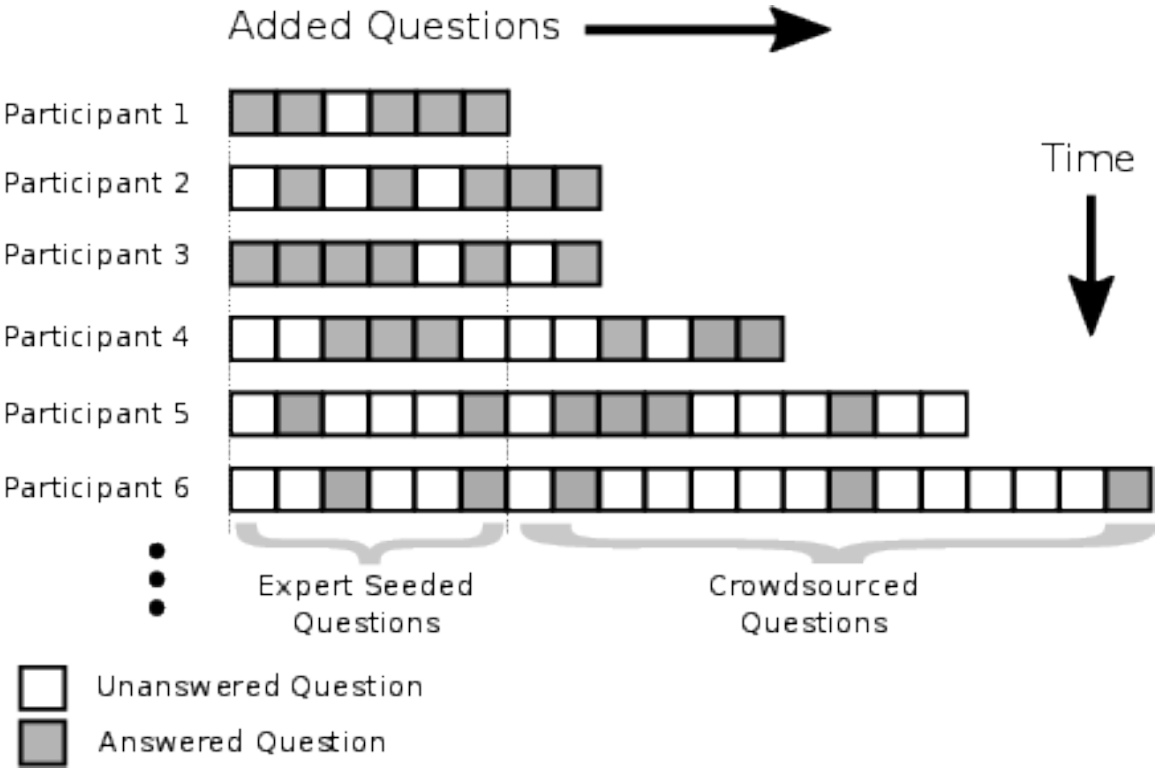

Figure 5.1: ***Crowdsourcing questions and answers.*** *The process begins with a set of seed questions for which the first participant contributes answers. The first participant then contributes two of their own questions. The second participant contributes answers to a sampling of the available questions, both seeded questions and questions contributed by the previous participant. This process continues over a set time period. As questions are added, the sparsity of the dataset grows because many of the questions contributed late in the process were not available to early participants. These questions go unanswered by earlier participants unless the participant returns to the site.*

Note that this type of question/answer feedback system necessarily results in a very sparse dataset. For example, consider the question posed by the last user of the system. This question would have only one answer associated with it since only the last user could answer it. Similarly, a question posed by the second-to-last user of the system would only not have more than two answers; and so on. What results from such a system is necessarily a sparse dataset of questions and answers.

5.2.2 Ex post data analysis

After the experiment concluded, we revisited the resulting dataset in order to better understand whether the crowd could contribute to a predictive model of energy usage and to identify modeling methods that can most effectively identify valid patterns in the uniqely sparse datasets that result from this approach. To do so we built new outcome variables $E_{90,i}$ from user energy consumption values during a time period with both high participation in the EnergyMinder tool and when there was high electricity demand: from December 21, 2013 to March 21, 2014. This outcome was then used, along with the user-contributed questions and answers to build a predictive model and identify user-contributed questions (alone and in combinations) that were predictive of energy usage. The resulting dataset, $\mathcal{D}$, related each participant in the study, i, to a user-contributed question, j. The presence of a value at $\mathcal{D}_{i,j}$ indicated that user i contributed an answer to question j.

We used a random forest regression algorithm [114, 115] to train the *ex post* predictive model. Random forest regression is an ensemble-based machine learning method [116], which consists of training a set of weak learning models on subsets of data using a process known as bagging. Bagging has been shown to be appropriate for building models on data that can, if perturbed, greatly alter the performance of the learned model and for 'data with many weak inputs' [117]. An overall classification or regression prediction is then obtained by aggregating the output of the weak learners to obtain a predictive model, $\mathcal{P}$.

The explanatory features being used to build the model – user-generated questions – consisted of both numeric- and categorical-valued data. Due to the large degree of

sparsity inherent in the process of collaborative crowdsourcing, our expectation was that the model fit would include a large amount of noise. However, we were mainly interested in whether some signal could be found in that noise, however slight. If $\mathcal{P}$ could outperform a model not incorporating user input; and it uses user-generated features despite the sparsity challenges, we could conclude that it had some value in explaining behaviors that contributed to residential energy usage. And it would thus demonstrate that the crowd indeed contributed to a predictive model.

The random forest regression algorithm requires that all values be present. To obtain a full dataset, we utilized the method of mean imputation. This imputation choice was governed by necessity: standard imputation methods, such as list-wise or pair-wise deletion would have resulted in a dramatic reduction in samples on which a predictive model could be built, rendering the modeling process impracticable. And due to the idiosyncrasies of this particular type of dataset, in which there is little overlap in answered values across features, methods that attempt to estimate joint distributions such as Bayesian multiple imputation methods [118] were not able to converge. After performing mean imputation, we normalized all values to z-scores.

To demonstrate that our model $\mathcal{P}$ could find some signal in the sparse dataset, $\mathcal{D}$, we compared it to a null model, $\mathcal{P}_{\text{null}}$. This null model was trained using the same random forest regression algorithm. However, it was trained on a randomized (shuffled)version of $\mathcal{D}$ (denoted $\mathcal{D}^{\text{shuf}}$). $\mathcal{D}^{\text{shuf}}$ was obtained by randomly reordering user answers for each user-contributed question (along the margin, $\mathcal{D}^{\text{shuf}}_{*,j}$). This had the effect of maintaining the basic statistical properties of $\mathcal{D}^{shuf}$ along the feature margins while disassociating the contributed answers with each user and their energy usage totals. If we were able to build predictive models $\mathcal{P}$ with consistently lower

error than $\mathcal{P}_{\text{null}}$, then we can conclude that the set of features used in the model are indeed predictive.

From random forests, we obtain a large set of decision trees whose leaves are aggregated to obtain a prediction for each set of inputs. The growth of such trees is governed by information reduction branching criteria on the available features. Despite the difficulty in interpreting a large collection of decision trees, which form a basis for random forests, we can obtain variable importance rankings [114]. We used the Gini impurity index to build a ranking of the features used in the model [119]. Thus if crowd-generated questions have high rankings, we can reasonably assume that the crowd did indeed contribute useful information to the model.

However, we do not expect that all features in this ranking are contributing meaningfully to the predictive model. That is, there is a point at which the ranking of features transition from those contributing to the regression model and those that are included only by chance. To differentiate between the features that are meaningful versus those that are included only by chance, we utilized a technique for estimating the random degeneration between lists [120]. In this method, we can obtain a cutoff point at which a set of ranked lists begin to diverge into random orderings [121].

To obtain error estimates for $\mathcal{P}$ and $\mathcal{P}_{\text{null}}$, we trained a set of 10 independent Random Forest Regression models. We used the 10 associated feature importance rankings for $\mathcal{P}$ for comparison. We compared these 10 lists to estimate a value k at which they began to deviate from a meaningful ranking of features that were consistently used in the model rather than being included only by chance. Those features that are ranked higher (i.e. lower in rank number) than this degeneration cutoff, k, are considered to be used in the predictive model not by chance, whereas those above

k were included due only to chance. As this method relies on parameters (ν and δ, see [120]), we performed a sensitivity study over a range of parameterizations.

5.3 Results

From the period in which EnergyMinder was deployed from June 25, 2013 until September 24, 2014 a total of 627 active crowd-participants (those who answered at least one question) asked 632 questions, and provided $110,573$ answers to questions. Of the 632 questions posed by the crowd, 627 were answered at least once. Figure 5.2 shows the pattern of missing values in the resulting dataset. In this plot users are ordered by the time that they signed up to participate in the study. The amount of sparsity in a question ranges from a maximum of 100% missing to a minimum of 32.1% missing.

Table 5.2 shows representative top questions in ranked order by the random forest regression method described in Section 7.3. The average mean squared error for the predictive model $\mathcal{P}$ was 0.883 and the average mean squared error for $\mathcal{P}_{\text{shuf}}$ was 1.031, which was significant at $p < 0.0001$ ($p = 0.00001083$; Kolmogorov-Smirnov test, $D = 1$).

The list-wise random deviation method for obtaining cutoffs in ranked lists (described in Section 7.3) resulted in cutoff values k that ranged between 9 (for $\delta = 1$ and $\nu = 2$) and 89 (for $\delta = 10$ and $\nu = 9$). Our sensitivity study over the parameters, δ and ν was run over a range of $[1, 10]$ for δ and $[2, 20]$ for ν.

We also ran the random deviation cutoff method on $\mathcal{D}_{null}$ to obtain a list of ranked user-contributed questions. We did this for the same range of parameters as

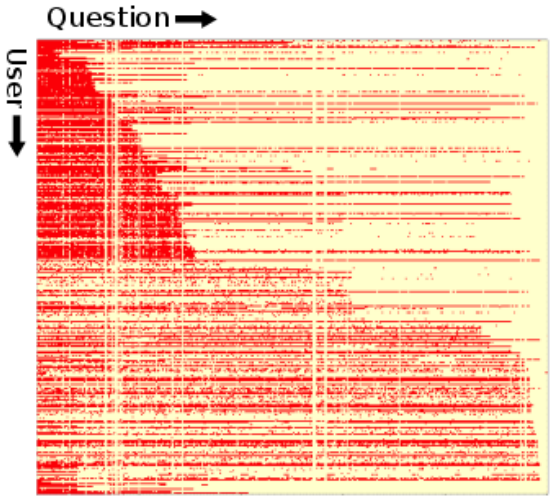

Figure 5.2: ***Missing value pattern plot.*** *Each red dot represents a user answering a question. Questions are shown along the horizontal dimension and users are on the vertical dimension. Total counts of answers per question are shown in the barplot in black. Note that users were given sequential id numbers such that customers that joined later received larger user id numbers.*

in the models trained on $\mathcal{D}$. Out of all possible parameterizations, 56% instances of running the algorithm were not able to find a valid cutoff value that differentiated the meaningful rank comparisons from the ranked features that were included only by chance. Of the times that the algorithm was able to find a valid cutoff in the question ranking, 56 cutoff values of 5, 22 cutoff values of 6, and 6 cutoff values of 7.

5.4 Discussion

5.4.1 Contributed Questions and Participation

The number of users providing answers (627) was very close to the number of questions in the system (632), thus creating a dataset that is approximately as 'wide' as it is

Table 5.2: Top 43 user-contributed questions as determined by random forest regression. Expert-created question IDs in bold.

Question ID	Question Text
q4	How many loads of laundry do you do per week?
q76	How many TVs are in your home?
q1	I generally use air conditioners on hot summer days
q24	How many hours of TV or DVD/Video viewing is there, in your home, per week?
q142	Do you use washer and dryers outside of your home?
q13	How large is your home in square feet of living space?
q335	I know most of my neighbors on a first name basis.
q109	Do you use heat tape during the winter on the pipes of your mobile home?
q34	How many months of the year do you use your dehumidifier?
q77	How many hours a day is there someone awake in your home?
q18	Do you have a dehumidifier?
q7	How many people live in your household?
q12	Do you have central air conditioning?
q22	How many pieces of toast do you toast on a typical week?
q62	How high (in feet) are the ceilings on the main floor of your home?
q86	Do you live in a rental unit that supplies your hot water heater?
q283	I'd rather invest in energy efficient appliances than save for retirement.
q8	I only run the dishwasher when it's full.
q6	Most of my appliances (laundry machines, refrigerator, etc.) are high efficiency.
q56	How many rooms in your home have an exterior wall and windows that face east?
q74	What's the size of your home or apartment, in square feet?
q167	Do you use solar powered exterior lighting?
q81	What temperature is your thermostat set at?
q54	Do you use a garbage-disposal unit in your sink?
q290	We should leave porchlights on as a courtesy to our neighbors.
q63	How high (in feet) are the ceilings on upper/additional floors of your home? (Answer 0 if not applicable)
q107	Do you live in a mobile home?
q30	How many cars are generally parked at your home each night?
q72	What percentage of your lights have dimmers?
q17	Most of my lighting is high efficiency CFLs or LEDs
q95	How long is your typical shower in minutes?
q2	Do you have a hot tub?
q9	I only use lighting when necessary.
q121	How many double paned windows are in your home
q11	How many room/window air conditioners are in your home?
q141	How many times a month do you eat away from your home?
q332	When presented with options for food sources, I usually buy local.
q3	How many teenagers are in your home?
q43	At what temperature is your electric hot water heater set?
q20	Do you have an electric oven?
q114	How many months per year do you hang clothes to dry?
q38	Do you have a microwave oven?
q25	My desktop computer is always on.

'long'. The median number of questions answered by a single user was 105 questions and the maximum number of questions that a single user answered was 585. A surprising number of users answered hundreds of questions (see Figure 5.3). A total of 166 users answered more than 100 questions. However, at least 32% sparsity is present in all of the features used in each of the models. Thus the most completely populated feature contains answers from only $\sim$ 68% of users. And only a subset of these overlap with answers in other features. Starting at the top of Fig. 5.2 and moving down to the bottom, we can see a widening band of answered questions starting as users increasingly participate. The questions outside of this band indicate

that users did indeed return to the study to answer questions that were posed after they had visited the study for the first time. At roughly the vertical middle of the figure we see a point at which a large number of questions were added by either a single or just a few users as indicated by the rapid increase of number of questions answered.

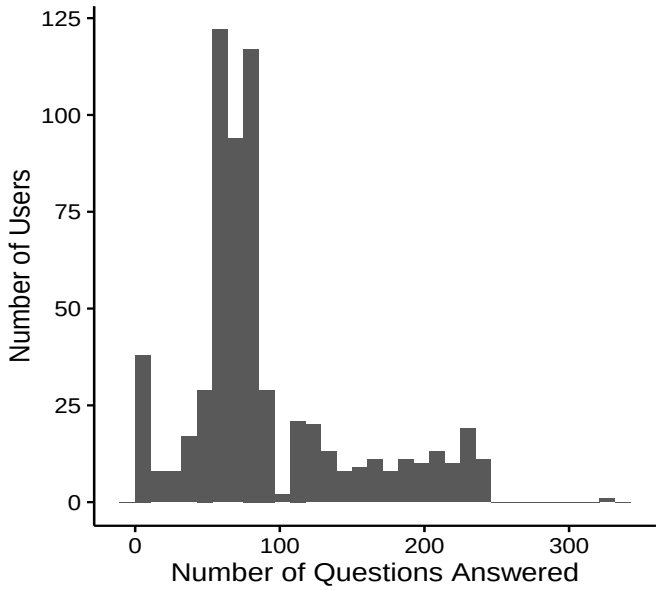

Figure 5.3: ***Histogram showing user participation.*** *A large number of users answered more than* 100 *questions in total.*

There are prominent vertical bands in Figure 5.2 indicating questions without any answers (or very few answers). While users were given the opportunity to skip a question, most of these empty questions can be explained by those questions that were rejected by the moderator for not following directions, being offensive or asking respondents to reveal too much personal information.

Figure 5.4 indicates the number of questions that were asked over time. The vast majority of questions were asked in a short period of time. This period corresponded

to a call for participation to BED customers that were received via mail.

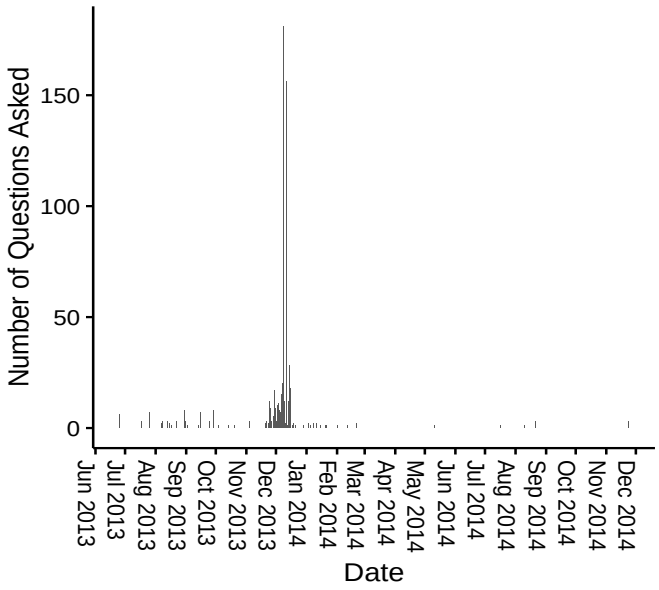

Figure 5.4: ***Histogram showing the number of questions asked over the study's active time period.*** *Questions were mainly asked in the December, January time period in the winter of 2013/2014.*

Figure 5.5 shows a list of questions whose correlations with the outcome – residential energy consumed – were greater than 0.15. The highest correlated question, *q4* ("How many loads of laundry do you do per week?") was one of the expert-seeded questions seen in Table 5.1. But the second and third highest correlated questions with the outcome – *q7* ("How many people live in your household?") and *q18* ("Do you have a dehumidifier?") – were asked by the 'crowd'.

Note that in the 18 questions with correlations greater than 0.15 to the outcome, only one question is negatively correlated. Out of all questions with a correlation of at least 0.01, only 16% were negatively correlated. That participants appear to focus on questions that are positively correlated with the outcome may be the result of priming, or it could be evidence of a cognitive bias.

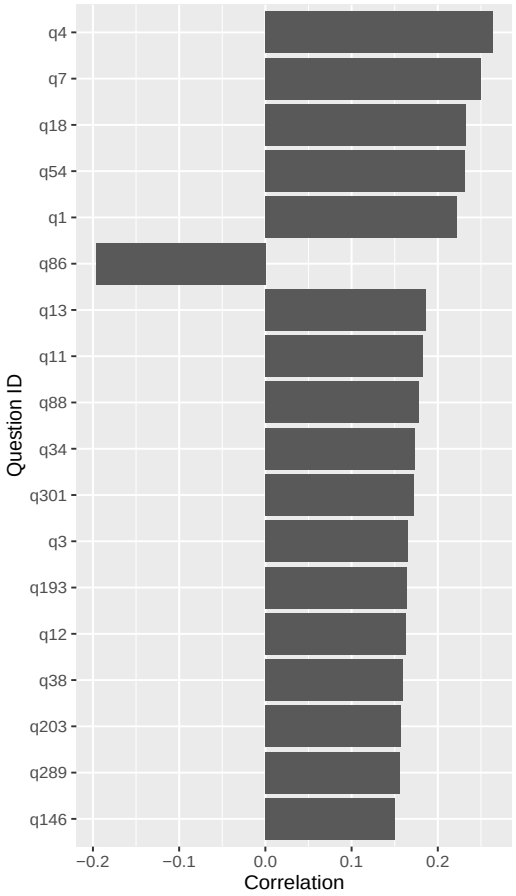

Figure 5.5: ***Correlations of questions with a correlation value greater than*** 0.15***.*** *Questions are ranked from highest absolute correlation value to lowest (top to bottom).*

5.4.2 Predictive Model

Our true model ($\mathcal{P}$) to null model comparison ($\mathcal{P}_{null}$) does result in a significant difference between the error the predictive model trained on the true dataset, $\mathcal{D}$. Therefore, the models trained using the random forest regression algorithm were able to find some 'signal in the noise'. And thus the questions did indeed provide some degree of predictive power in building the models.

It may be that only the expert provided questions ($q1$ - $q6$) contributed to the

predictive ability of $\mathcal{P}$. If this were the case, we would not be able to say that crowd hypotheses contributed to a predictive model.

However, the analysis of the random degeneration of lists indicates that a large number of features used between models are not included by chance alone (on average, the top 51 ranked questions from $\mathcal{P}$ were considered not to be random between models, see Figure 5.6). This is in contrast to the null models generated, in which the majority of parameterizations resulted in no valid point at which the feature rankings deviated from meaningful to random collections.

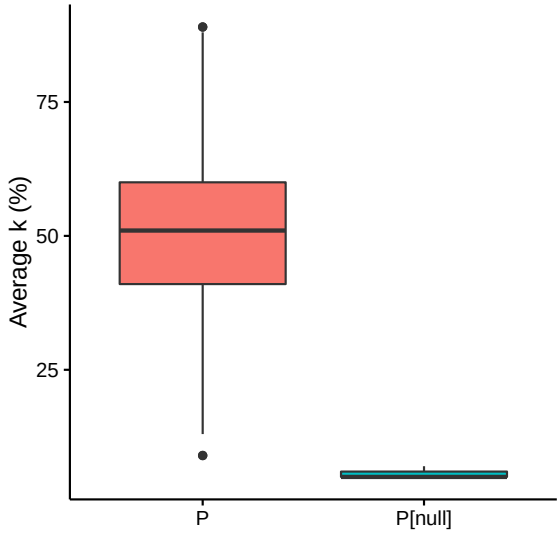

Figure 5.6: ***Number of top questions used in building $\mathcal{P}$ versus the null model $\mathcal{P}_{null}$ over all parameterizations of ν and δ.*** *The mode value for the number of top ranked questions in $\mathcal{P}$ was* 43 *(median value of* 51*), whereas most of the attempts at finding a common cutoff for degeneration, k, failed in the question rankings derived from $\mathcal{P}_{null}$.*

The questions in the list of top ranked questions (Table 5.2) can be broadly classified into addressing energy choices, lifestyle and behavior (e.g., *q1, q4, q24, q77*); appliances and house features or layout (e.g., *q18, q13, q62, q167*); and house inhabitants (e.g., *q3, q7*).

Two of the top 10 questions were expert-generated questions. The probability of two expert questions being found by chance in the top ten is $\binom{6}{2} \times \binom{626}{8} \div \binom{632}{10} \approx 0.0008$. Thus we have reasonably high confidence that the model incorporates questions that are useful in its top-ranked questions if we are to assume that the expert is able to formulate predictive questions.

Some of the ways that questions contributed cannot be explicitly measured by the models that we have built. For example, whether an earlier user-generated question had an impact on subsequent, possibly more predictive questions, is not readily apparent. For example, a member of the crowd asked the question *q64*, "Do you generally watch TV in bed at night?". This may have prompted another user to ask question *q76*, "How many TVs are in your home?", which was a high-ranked question in the random forest model. Similarly, the expert-contributed question *q1*, "I generally use air conditioners on hot summer days" may have inspired the non-expert user-generated questions *q11* and *q12*, "How many room/window air conditioners are in your home?" and "Do you have central air conditioning?", respectively. The effect of social influence when asking questions and answering them may have had a positive effect on the overall ability to find predictive behaviors or it could have just as easily negatively influenced the overall crowd effort to explain energy usage. For example, social influence could have caused members of the crowd to become trapped in group pathologies such as groupthink [38]. Details on how the crowd mutually influenced each other is out of the scope of this work, but would be one direction to explore in future work.

Note that some of the predictive questions relate energy consumption to air conditioner use, yet the energy usage data that we are using for training the models is

based on data from the winter months. This may at first appear counter-intuitive. It is possible that questions like this uncover general trends in behavior. For example, someone with a low tolerance for discomfort in the summer months also has a low tolerance for discomfort in the winter months. Thus, being the type of person who uses electricity for air conditioners during the summer could be indicative of the same tendency to use heaters that are energy sinks in the winter.

Most of the highest ranked questions appear to related directly to energy usage or behaviors that might clearly affect household energy consumption. However, the seventh-highest-ranked question (*q335*) appears unrelated to energy consumption: "I know most of my neighbors on a first name basis." We can only speculate as to why this question might be predictive of the outcome – or why the participant who asked the question believed that it would be predictive. That this question was asked and found to be predictive indicates that there may be evidence of a relationship between a person's connection with their neighbors and their energy usage. It is interesting to note that that participants were exposed a graph indicating their own energy usage compared with other participants in the Energy Minder interface. The question may have been influenced by this feature of the website.

Many of the questions ranked highly by random forests were also highly correlated with the outcome (Figure 5.7). For example, *q4* ("How many loads of laundry do you do per week?") was most correlated to energy usage and was ranked highest by the random forest algorithm's mean Gini decrease importance criterion. Also, *q1* ("I generally use air conditioners on hot summer days"), *q13* ("How large is your home in square feet of living space?"), and *34* ("How many months of the year do you use your dehumidifier?") all appear within the top 10 ranked questions both measured

with respect to correlation to the outcome and importance calculated from random forest regression. But there were some instances of questions that were deemed more important by random forest regression than in direct correlation with the outcome. In particular *q142* is ranked 5^{th} by random forest regression importance measures and is ranked near last (595^{th}) by its correlation to the outcome. Random forests are non-linear regression algorithms. Thus there may be non-linear relationships between such questions as *q142* and energy usage that would not be discovered solely through the use of linear models.

5.5 Conclusion and Future Work

This paper presents a new method for using crowdsourced models to explain why some homes use more or less electricity than others. This approach uses random forest regression to relate answers to questions posed by the crowd participants themselves, to predict and identify key factors that influence residential electric energy usage. Because of the crowd-driven nature of the data collection process, the data were necessarily sparse. This is due to the nature of survey participants both generating the features that the models fit by contributing questions to a survey and by contributing data to these questions by answering them. The sparsity in the dataset is nonuniform due to increasing numbers of questions becoming available as the result of user participation, which then go unanswered by previous participants.

The predictive model resulted in significantly less error than a model trained on a randomized version of the same data. This indicates that the predictive model is indeed finding behavioral drivers of energy consumption. Additionally, we found a

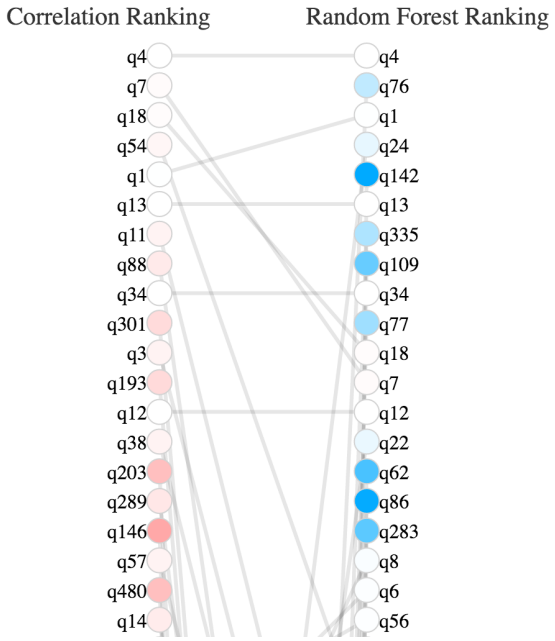

Figure 5.7: ***Rankings of (top** 20**) user-contributed questions by correlation with energy usage value (left column) compared to the rankings of questions computed via the mean Gini impurity ranking method used for importance ranking in the random forest regression algorithm (right column).*** *Top ranked question is at the top (q4), the second ranked question by each method is the second from the top (q7 for correlation and q76 for random forest regression), and so on. Lines connect questions between rankings. The amount of fill indicates the amount of difference in ranking from one column to another. Red fill in a circle associated with a question indicates a decrease in ranking from left to right and blue for an increase in ranking from left to right. The amount of white in a circle's fill indicates a neutral change in ranking.*

cutoff that differentiated those questions that were used by the model in a meaningful versus random way. And this cutoff necessarily incorporated user-contributed questions. Therefore the crowd contributed in a significant way to the models. The questions incorporated into this models give us clues as to what types of behavior are important to residential energy usage.

While some of these contributed questions were generated by expert users (though also members of the population of residential energy users), some of them were pro-

duced by non-expert members of 'the crowd'. Thus the intuition of the crowd can be used to develop models for residential energy consumption, and possibly other domains.

There are several opportunities for further work with this methodology for building predictive models from collaborative crowdsourced data. Future work should address whether increasing the number of questions and the number of users improves regression fit and classification accuracy even though the proportion of sparsity stays the same, i.e., whether a 'longer' and 'wider' dataset, but with the same amount of sparsity (i.e., an even larger study) proves beneficial or detrimental to the predictive model.

Additionally, there are opportunities to pose user-generated questions in an adaptive way. For example, it may be possible to motivate the crowd to pose questions that are most likely to attract answers and thus even more explanatory new questions.

This collaborative crowdsourcing method may also be used as a complement to geographically specific load forecasting methods. The results of this study suggest that – in addition to using standard expert opinion and historical load data – we may be able to complement existing predictive model features with consumer feedback to potentially enhance the accuracy of load forecasts where advanced metering infrastructure is available.

Chapter 6

Crowdseeding with Observable Features*

Abstract

Crowdsourcing is a well-known method in which intelligence tasks are completed by an anonymous group of human participants. These are tasks that cannot yet be adequately performed by machine intelligence methods. As crowdsourcing necessitates human-computer interaction to complete tasks, various strategies exist to partition labor between the human participant and the computer to effectively solve problems. One such crowdsourcing strategy is one in which human intelligence is used to complement machine intelligence tasks rather than perform them outright. A key point in determining the potential of such a strategy is understanding the ways that human abilities most effectively complement the strengths of machine intelligence.

We shed light on this relationship by using results from a web-based experiment in which robot design was crowdsourced to anonymous, non-expert volunteers. We investigate the effectiveness with which human-devised design strategies can be incorporated into a learning strategy as an additional objective in a genetic algorithm. We

*Appeared as (extended abstract) Wagy, Mark D., and Bongard, Josh C. "Crowdseeding robot design." Proceedings of the Companion Publication of the 2015 on Genetic and Evolutionary Computation Conference. ACM, 2015.

demonstrate that by using an objective parameter in this machine learning algorithm which was seeded by the efforts of human participants, we are able to outperform the same algorithm without the human contribution.

6.1 Introduction

With the popularity of crowdsourcing [7] on the rise, increasing effort is being channeled into understanding how we can best combine distributed human intelligence tasks effectively to perform more and more complex tasks. Crowdsourcing, and more generally human computation [9], necessarily involves some level of shared work between a human or a collective of humans and a computer or group of computers. A significant challenge lies in understanding what role each of the human and computer components of this interaction should play to best complement each others' characteristic abilities. Exploring novel and effective techniques for combining the abilities of humans with the capabilities of machines will allow us to discover the ways that we can use the the wisdom of the crowd [122] to solve more complex problems.

In the present study we use features extracted from a crowdsourced group activity – that of designing robot morphologies that are able to locomote effectively – to seed a machine learning algorithm with the aim of making it more effective at its task. When considered broadly, this form of interaction is common: ideas are developed by interacting groups of people that are then translated into a computer process. In this study, we observe common tendencies in the group activity, which the participants may or may not be consciously aware of. We then use these tendencies to direct a computational evolutionary process. We suggest that the tendencies favored by the human group relates to an advantage that they have over the machine learning algorithm: that of being situated in a physical environment. This is used as a complementary advantage to the computational efficiencies of machines.

6.2 Related Work

Crowdsourcing is by now a well-established means for completing a series of tasks considered simple by human standards but beyond the reach of machine intelligence [9]. But interest in crowdsourcing is moving beyond using it to complete simple tasks and towards understanding how we can combine the abilities of humans to solve more complex and creative problems that that transcend the abilities of any one participant [123]. Substantial successes in utilizing crowdsourcing for solving complex problems has been demonstrated in such diverse areas as classifing galaxies [15], discovering protein folding algorithms [14] and decision-making for disaster relief [124]. Crowdsourcing has been used to improve grasping behavior in robots [72] and influence the way robots interact socially with humans [71]. Using so-called *games with a purpose* [17] studies have show that [11, 125] we can extract useful information through engaging user interfaces even if the user may be unaware that they are doing more than simply playing a game. In addition to demonstrating the effectiveness of the crowd at solving complex domain problems, these studies demonstrate the important relationship between the human crowd and the role of the computer in extracting behavior conducive to solving problems collectively.

Rather than simply serving as an interface through which the crowd can interact to solve problems independently, the computer component in this human-machine interactive system can play a more active role. In [126], machine learning and planning algorithms were used to guide the hiring of participants in a crowdsourced task, thus actively directing the solution of the problem towards a more accurate one by deciding which of the human contributor's input is considered. Studies in inter-

active evolutionary computation incorporate human input into an evolution-based machine learning algorithm in using humans to guide selection based on evaluation of expressed phenotypic characteristics [25, 127] and to contribute to the process of recombination [27] of genotypes in the population. In [69], the crowd worked with an evolutionary algorithm to guide evolutionary search for robot controllers using a web-based interface. [44] and [43] use human input to guide search for improved robot control strategies.

Whereas in past work individuals actively contributed to selection and mutation operations in an evolutionary algorithm, in the present study we use traits distilled from a creative, crowdsourced activity to supplement the objective of evolutionary search; namely that of a search for robot morphologies that result in effective locomotion. This is similar in some respects to the discovery of common sense rules from crowdsourced activities to build a machine intelligence model [28–31]. However, in contrast to these studies we are leveraging implicit ideas extracted from a crowd activity to seed a machine learning algorithm with the goal of improving its performance in a creative task rather than using common sense to build a stand-alone decision model.

6.3 Methods

The experiment consisted of two stages. In the first stage of the experiment, we deployed a web-based interface (Section 6.3.1) in which users were able to design robot bodies using a simple grid-based drawing tool. In the second stage, we used prominent features observed in these morphologies favored by participants as an additional

objective in a genetic algorithm (Section 6.3.2) to the primary objective of moving as far as possible from its initial point.

6.3.1 Stage 1

User Interface

Users were invited to design robots using a web-based user interface. The interface consisted of three main components: a *design panel*, a *history panel* and a *simulation panel* (Figure 7.1). They used the design panel (Figure 7.1C) to draw a robot. Then when the user clicked the *GO* button, the participant was able to see their robot design move in the simulation panel (Figure 7.1B). They were able to track the distance that the robot moved both visually and via distance indicator above the panel. They were also shown a history panel (Figure 7.1A), which contained a random sample of robots designed by other users along with the distance that that design was able to move. Users were free to use these other designs as guidance in their own designs or they could ignore these historical designs if they so desired.

The design panel consisted of a 5×5 grid of dots that users could connect by dragging their mouse pointer from one dot to another. The drawings were limited to lines between adjacent dots. Each of the drawn lines would translate to a $1 \times 1 \times 0.1$ meter segment in the simulated robot. Each dot that was adjacent to a line translated to a $0.2 \times 0.2 \times 0.2$ meter cube that was attached to adjacent segments with a rotational hinge joint. The axis of rotation of this joint was defined to be perpendicular to the normal of the ground plane and the normal of the object faces being connected. This axis of rotation allowed for pushing movement away from the ground plane. Robots

were simulated in the three-dimensional web-based physics engine, Ammo.js[†], and were rendered using WebGL[‡] in the participant's web browser.

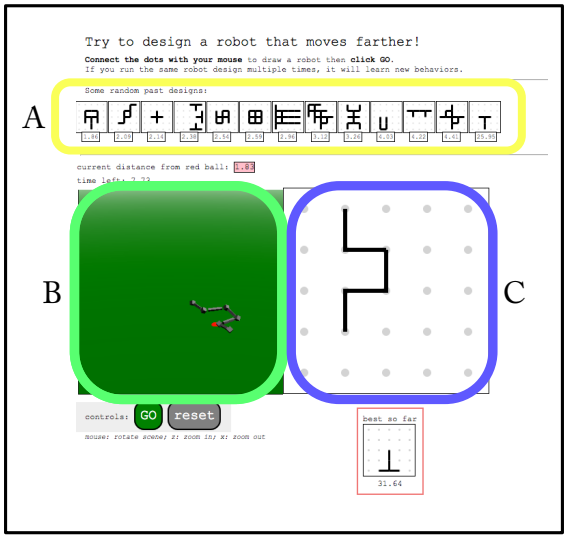

Figure 6.1: ***Screenshot of user interface.*** *Users could see a sample of designs produced by other participants at the top (A, yellow box). They could "connect the dots" to draw robot morphologies in a design panel on the right (C, blue box) and when they clicked on GO, they would see a simulation of the robot run in the simulation panel (B, green box).*

The robot was allowed to move in the simulation for 15 seconds along a flat ground plane. Each hinge joint in the robot was actuated via sinusoidal displacement-controlled motors. This hinge joints would sweep an angle of $[-45^o, +45^o]$ either in-phase or 180^o out-of-phase with each other thus allowing symmetric or antisymmetric actuation of joint motors.

When participants visited the site in their web browsers, they were only told that they should design a robot that moves as far as possible. They were not given any instruction as to how to accomplish this task. Participants were anonymous and unpaid.

[†]https://github.com/kripken/ammo.js/
[‡]https://www.khronos.org/webgl/

Hillclimber

Each time a user created a new robot body design, the two hinges associated with each segment would be randomly assigned to move in-phase (phase offset of 0^o) or out-of-phase (phase offset of 180^o) with other hinge joints. A separate hill-climber was maintained for each morphology to define the phase offsets of its joints so that if users within the same group were to redraw the same morphology, the hinge phase offsets would be chosen according to that morphology's hillclimber. As such, the "control" of the robot body could improve over repeated iterations of the same morphology, enabling participants to test their intuitions for what makes a good robot.

6.3.2 Stage 2

Evolutionary Algorithm

After the initial crowdsourced portion of the segment was complete in which human users designed robots, a genetic algorithm was used to evolve new robot bodies and their controllers. The genomes in this genetic algorithm consisted of bitstrings. Within the genome, each set of three consecutive bits represented a potential segment in the robot. The center bit of each grouping of three bits represented either the presence or absence of a segment at each location in a 5×5 grid. If the middle bit in a grouping of three was zero, the left and right bits were ignored. If the middle bit in a grouping was 1 (indicating the presence of a segment), the left and right bits surrounding it represented either in-phase (0) or 180^o out-of-phase (1) actuation of the segment's connecting hinge joint (see Figure 7.2). For each possible segment in the grid there were three such groupings; as such, each genome consisted

of $5 \times 4 \times 2 \times 3 = 120$ bits.

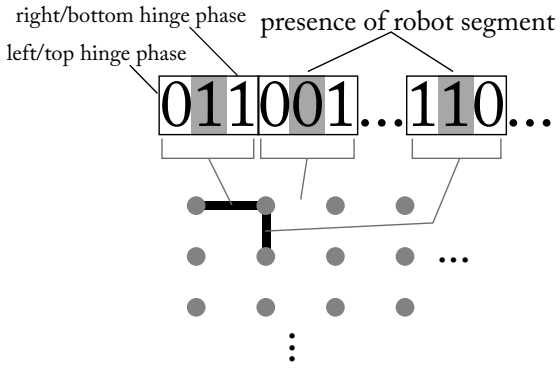

Figure 6.2: ***Genotype to phenotype relationship.*** *Whether a segment is drawn between two dots is represented by the middle bit in each grouping of three bits. The bits to the left and right of it control whether the hinge is actuated to oscillate at* 0^o *phase offset or* 180^o *phase offset.*

Each experimental sample consisted of running the genetic algorithm for 100 generations. We used bit-flip mutation at a rate of 0.1 as well as uniform crossover at the same 0.1 rate. The fitness of each genotype was determined according to the objectives defined in Section 6.3.2. The experimental treatments employed multiobjective selection in which non-dominated individuals on the multiobjective Pareto front were selected to survive to the next generation.

We performed 100 independent evolutionary runs of the control case to compare with 100 independent runs of the first experimental treatment and another 100 independent evolutionary runs of the control treatment to compare with 100 independent runs of the second experimental treatment.

Objectives

In the control treatment, fitness consisted of a single objective: to maximize the distance a particular robot was able to move away from its initial position in the

physics simulation. For multiple component robots, distance was calculated according to how far the center of mass of all pieces was able to move away from its initial position. The experimental treatment had the same objective, but in addition to the distance objective, a second objective was selected for.

We explored two variations for the additional objective in our multiobjective experimental treatments. These additional objectives in the experimental treatments were driven by prominent characteristics extracted from user preference in the first, user-interactive stage of the experiment. The first characteristic was that of symmetry in the robot bodies. We defined symmetry as the proportion of segments that are matched when the robot design is reflected about either the horizontal, vertical or diagonal lines of the minimum bounding box that can contain the robot design in the plane of the 5×5 grid that is used to draw the morphology. The second characteristic was user preference for the minimization of components that make up the robot's body - that is, how many separate pieces the robot consisted of. In this second variation of the multiobjective treatment, in the fitness function we combined the maximization of symmetry with the minimization of number of connected components that made up the robot body (Table 6.1).

Treatment	**Objective 1**	**Objective 2**
Control	distance (+)	—
Experimental I	distance (+)	symmetry (+)
Experimental II	distance (+)	symmetry (+), components (−)

Table 6.1: Experimental treatments. (+) *denotes that the objective is to be maximized,* (−) *denotes that an objective is to be minimized.*

Total evaluations	11800
Number of participants	947
Number of unique designs	2292
Designs per participant	5.63 ± 7.13
Number of perfectly symmetric designs	1549
Number of singly-connected components	2211
Symmetric designs (of best 100)	79
Single-component designs (of best 100)	99

Table 6.2: User participation. Each time a user designs a robot body and presses GO in the web interface is considered an evaluation.

6.4 Results

Some examples of robot morphologies that users created in the first stage of the experiment can be seen in Figure 6.3 and numbers indicating participation in the first stage of the experiment can be seen in Table 6.2.

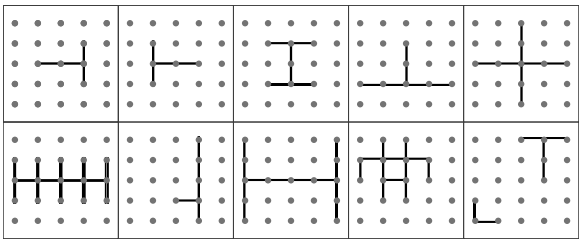

Figure 6.3: ***Ten examples of user designs of robot bodies.*** *Users could connect dots (gray) to "draw" segments of a robot (seen from the "top-down" perspective). Of the 100 robots that moved the furthest 79 were completely symmetric with respect to either a horizontal, vertical or diagonal axis. Of these 100 robots, only one consisted of more than one component.*

In addition to the numbers indicated in the Table 6.2, we can see the amount of symmetry and the numbers of components that users tended to favor in the first stage of the experiment in Figure 6.4.

The best distances achieved in the second stage of the experiment can be seen in

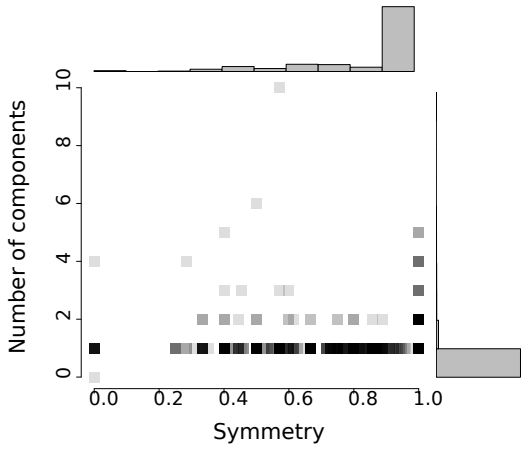

Figure 6.4: ***Number of components in all user designs in the first stage of the experiment and amount of symmetry in those designs.***

Figures 6.5 and 6.6 for the control cases and both of the experimental treatments. The mean value of best distances in all independent samples over the generations of the evolutionary algorithm can be seen in Figures 6.7 and 6.8.

6.5 Discussion and Conclusions

Participants showed a clear presence for symmetry in designs and for those robot designs that consist of a single component. However, there are clear exceptions to this - in some instances, users simply drew what may have been "interesting" or amusing designs. When we incorporated the preferences for symmetry and both symmetry and a small number of components, we can see that the incorporation of human preference into the objectives resulted in robots that were able to move farther than those in which only the single distance objective was selected for. This is despite the reduced selection pressure on the original distance objective due to multi-objective selection.

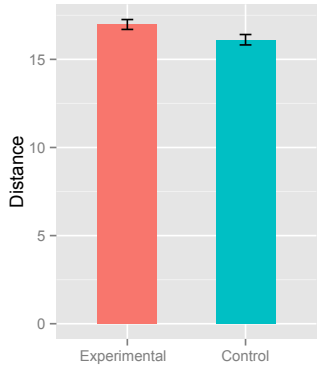

Figure 6.5: ***Best distance of control treatment versus experimental treatment I at the end of the evolutionary run (Mann-Whitney U-test, p=0.03804).***

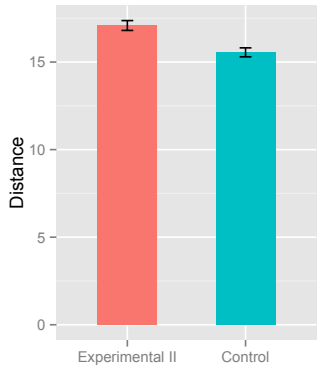

Figure 6.6: ***Best distance of control treatment versus experimental treatment II at the end of the evolutionary run (Mann-Whitney U-test, p=0.00017).***

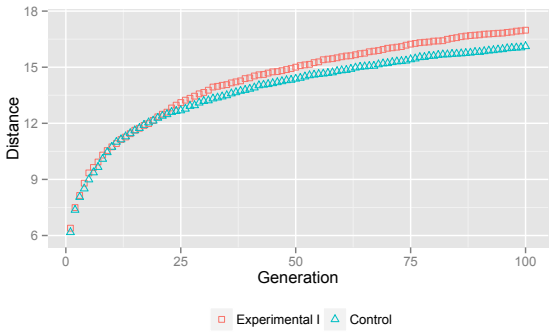

Figure 6.7: ***Comparison between control trials with experimental treatment I (distance objective and symmetry objective).*** *Mean value of the best distance achieved for independent trials over 100 generations.*

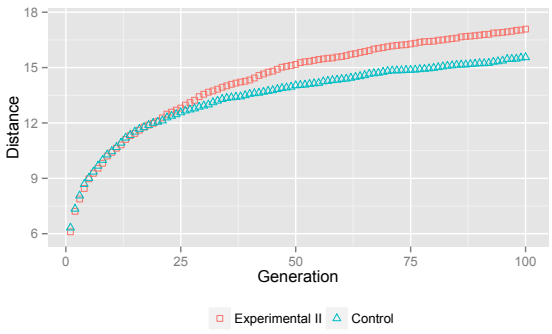

Figure 6.8: ***Comparison between control trials with experimental treatment II (distance objective and symmetry/number of components combined objective).*** *Mean value of the best distance achieved for independent trials over 100 generations.*

In the first experimental treatment, we incorporated symmetry as a second objective and found that the best distances achieved when incorporating both symmetry and distance objective outperforms uni-objective selection based solely on distance (Mann-Whitney U-test; $p = 0.03804$). Study participants were not given any guidance on how to design robots yet they showed a (possible subconscious) proclivity for designing symmetric robot morphologies, which corroborates current scientific understanding of the efficiency of symmetric morphologies in robot locomotion [101]. It is worth noting that the designs created by the genetic algorithm do not qualitatively resemble those that were used in determine how to seed the genetic algorithm through objectives (e.g. see Figure 6.9). We hypothesize that this understanding comes from their experience with animal locomotion in the natural world, which they in turn used to contribute to the design of robot morphology in ways that they may or may not be aware.

Similarly, we incorporated two user design tendencies into the second objective and we found significant improvement of the human-seeded evolutionary runs in this case (Mann-Whitney U-test; $p = 0.00017$). We incorporated symmetry in the second objective but additionally incorporated the preference for designing robots with a minimum number of components. Again, we did not instruct participants that they should build singly-connected robots; but the vast majority of users did so, using common sense and experience (although some users did indeed create multi-component robots).

Multi-component robots are a natural expression of the genotypic representation in the evolutionary algorithm used in the second stage of the experiment and symmetry is not necessarily characteristic to its representation (see Figure 6.9 for some examples

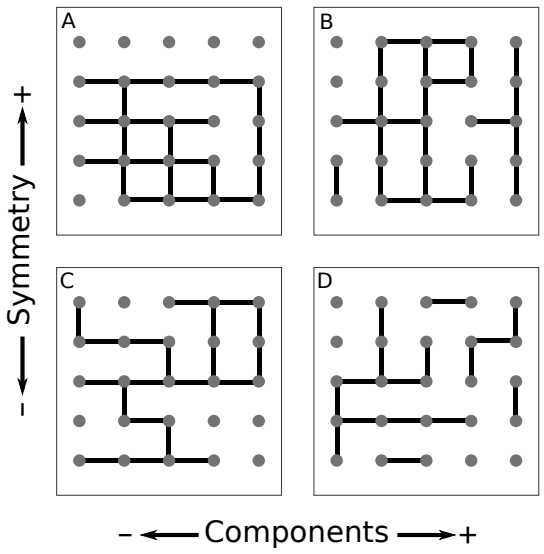

Figure 6.9: ***Four example robot morphologies generated using the genetic algorithm representation with varying amounts of symmetry and numbers of components.*** *Design A has a single component and a symmetry score of* 0.84*; design B has* 3 *components and a symmetry score of* 0.9*; design C has a single component and a symmetry score of* 0.5*; design D has five components and a symmetry score of* 0.4*.*

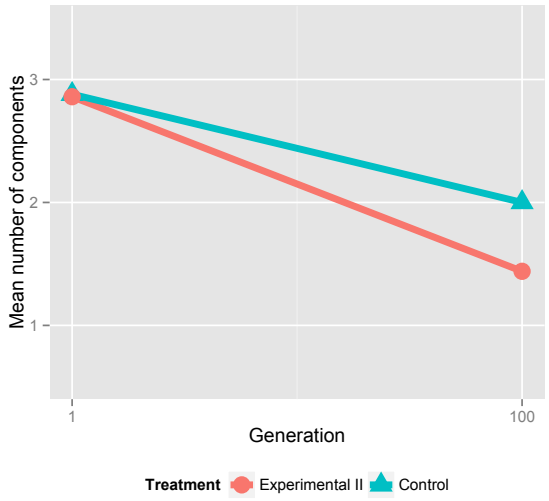

Figure 6.10: ***Mean value of number of components at generations 1 and 100 indicating typical amounts of symmetry in robot phenotype before the effect of evolution and after.***

of morphologies generated from the genetic algorithm representation with varying degrees of symmetry and numbers of components). We can see in Figures 6.10 and 6.11 that, on average, robot morphologies that have not undergone evolution (in the first generation) have approximatly 3 components and symmetry score of near 0.6 versus at the end of the evolutionary run in which symmetry and a smaller number of components are more common (obviously in the case in which they are selected for, but also in the control treatments). It was human intuition that limited the search space of the genetic algorithm to focus on the higher-performing subspace of single-component robot morphologies and symmetric robot bodies.

In the present study, we showed that seeding an evolutionary algorithm by incorporating user tendencies as an additional objective resulted in better overall performance with respect to the original objective. We did this by crowdsourcing robot morphology design and then incorporating features of the users' best designs into a genetic

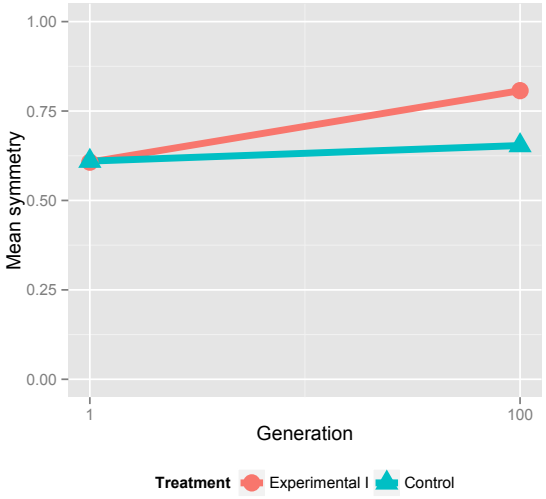

Figure 6.11: ***Mean value of symmetry measure at generations 1 and 100 indicating typical amounts of symmetry in robot phenotype before the effect of evolution and after.***

algorithm. One such preference was for symmetric designs and another for minimizing connected components. Both symmetry and a combined symmetry/component objective outperformed genetic algorithms selected for distance alone.

6.6 Future Work

We used user preferences to seed a genetic algorithm with the goal of designing locomoting robots in this study. Future work will examine other potentially less obvious features that users prefer using automated methods for feature selection and novel ways in which these features might be combined to make a superior additional seeded objective. This could result in unexpected ways in which robot morphology can be explicitly guided through evolutionary selection to produce effective robots.

Additionally, in future work we could use alternate representations for evolution

more suited to the crowdseeded objectives such as [128] for evolving symmetric bodies.

Chapter 7

Crowdseeding with Automated Feature Discovery*

Abstract

Crowdsourcing is a popular technique for distributing tasks to a group of anonymous workers over the web. Similarly, crowdseeding is any mechanism that extracts knowledge from the crowd, and then uses that knowledge to guide an automated process. Here we demonstrate a method that automatically distills features from a set of robot body plans designed by the crowd, and then uses those features to guide the automated design of robot body plans and controllers. This approach outperforms past work in which one feature was detected and distilled manually. This provides evidence that the crowd collectively possesses intuitions about the biomechanical advantages of certain body plans; we hypothesize that these intuitions derive from their experiences with biological organisms.

*Appeared as Wagy, Mark D., and Bongard, Josh C. "Crowdseeding: a novel approach for designing bioinspired machines." Biomimetic and Biohybrid Systems. Springer International Publishing, 2015.

7.1 Introduction

Embodied cognition [4] is the view that the intelligent behavior of an animal or human is influenced not just by its nervous system but also by its body plan. Engineers that produce bio-inspired designs are (implicitly or explicitly) adhering to this view. Wings on an airplane strongly suggest the influence of the morphology of birds. Many robots that have been developed are either humanoid in form [129–132] or resemble other animals, such as the canine *Bigdog* [133], the serpentine *OT-4* [134] or the chelonian *Aqua* [135]. Some biomimetic designs result from an explicit aim to exploit some desirable property of the behavior or feature of animals or of their environment. But in some cases the tendency to bias search toward specific design spaces could be considered an implicit tendency of collective human design behavior.

In [98], web participants collectively designed robot bodies. It was found that, among the successful designs, there was an overrepresentation of symmetric designs. This suggests that contributors have a strong proclivity for locomoting agents that resemble animals found in the physical world. Symmetry and single-component designs were the most explicit biases in robot bodies created by the crowd. However, there could be other, less obvious, traits and relations between them that could be exploited. In this study, we describe a novel methodology for extracting latent information from a group of participants in a crowdsourced experiment. Using the design of robot bodies and control as our domain, we use symbolic regression to discover implicit relations between properties of robot designs and an objective, in our case rapid forward locomotion. We then use these latent relationships to seed a new robot design process. This methodology represents a new mode of collaborative interaction

between a crowd of human designers and a machine learning algorithm.

7.2 Related Work

There has been of late a great deal of interest in finding methods to utilize the collective intelligence of crowds to solve complex problems [14,122,124,126]. The field of crowdsourcing has moved from being a convenient way to source simple, separable human intelligence tasks [136] that cannot yet be completed by machine intelligence [9] to being used to combine the efforts of individuals to solve larger problems that might not be amenable to reduction into a divide-and-conquer strategy [17, 137–139].

Use of human participants in evolutionary algorithms has been common for both selection of individuals in a population [25, 127] and in introducing variation in the evolutionary population [27]. But each of these examples involved direct user participation in the evolutionary search. Crowdsourcing in evolutionary robotics has been used to guide search for better robots and robot control [43, 44, 69]; but in these studies, the use of human intuition was used to actively guide search during the experiment rather than to distill out useful features in a crowdsourced study that were then incorporated into a separate search algorithm. In [98], features were extracted to seed the fitness objective of an evolutionary algorithm. But instead of using automated methods to find latent variables and their relations that contribute to performance – as is the case in this study – obvious characteristics of robot bodies favored by the crowd were distilled manually to seed the objective function.

7.3 Methods

We conducted an experiment in three stages. In the first stage of the experiment (Section 7.3.1), we deployed a web-based tool in which participants in a crowdsourced study designed robots collaboratively. In the second stage (Section 7.3.2), we used symbolic regression via genetic programming [140] to identify a novel relationship between the attributes of the crowdsourced robot designs and the distance that robots were able to move. In the third stage (Section 7.3.3), we used the relationship found through symbolic regression to augment a stochastic search process from a single-objective search problem to that of a multiobjective search.

7.3.1 First Stage: Crowdsourcing

We deployed a web-based tool that allowed participants to rapidly design robots using a simple grid-based drawing panel (Figure 7.1). We invited participants to design robots by recruiting them through the online forum *Reddit* (*www.reddit.com*). Participation was unpaid and voluntary. Participants were only given the instructions to design a robot that could "move farther". They were asked to connect dots to design a robot and click *GO*. They were told that their robot will learn new behaviors if they run the same robot multiple times.

Users connected dots in the design panel by clicking on a dot and dragging their mouse to another dot, which would form a line. Only lines between adjacent dots were allowed. When they clicked *GO*, each line was translated into a $0.1 \times 0.1 \times 0.1$ meter rigid segment in the simulation panel and each dot that was adjacent to a line was translated into a $0.2 \times 0.2 \times 0.2$ rigid cube. Cubes were connected to segments by a one-

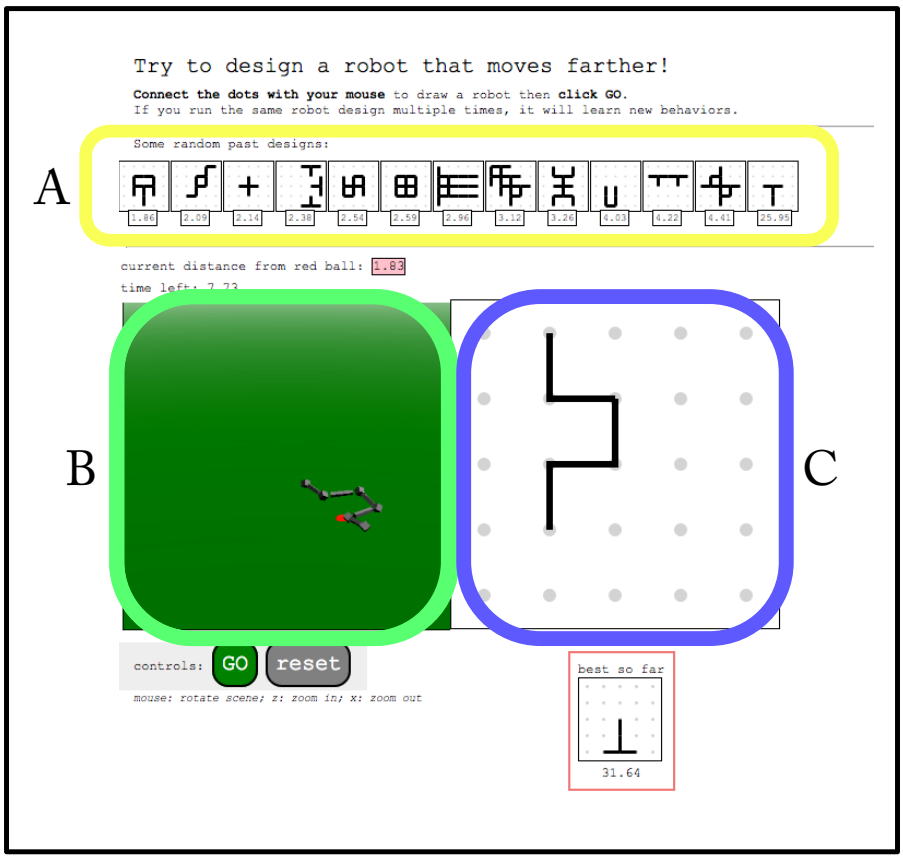

Figure 7.1: ***Screenshot of web-based robot design tool.*** *Users designed robot bodies by "connecting the dots" (Panel* ***C****). When they clicked "GO", they would see their robot move in a simulation (Panel* ***B****). They were shown a random sampling of 13 robots designed by others (Panel* ***A****).*

degree-of-freedom hinge joint. Robots were simulated in the three-dimensional physics simulation engine, Ammo.js[†], and were rendered using Web3D[‡] in the participant's web browser.

Each of the robot's joints was assigned to move either in-phase (0^o) or out-of-phase (180^o) with other joints. When a particular robot design was run for the first time, it was assigned its own hill-climber algorithm on a central repository,

[†]www.github.com/kripken/ammo.js
[‡]www.khronos.org/webgl

which would determine whether each of its joints would be in-phase or out-of-phase. If the same or another user repeated that design for a run in the simulation, the joint configuration would either be repeated or a joint could be randomly mutated from in-phase to out-of-phase or vice-versa at a 0.1 mutation rate. Thus every time a participant clicked *GO*, it was contributing one run to the hillclimber for that particular robot morphology. All joints were actuated with displacement-controlled motors via a sinusoidal signal with a frequency of 1.5 Hz, and sweeping an angle of $[-45^o, +45^o]$. The axis of rotation was defined to be perpindicular to the normal of the ground plane and the normal of the faces of the cube and segment being connected. Each time a robot was simulated, it was allowed to run for 15 seconds of simulation time. The distance that the robot moved from its starting point was displayed in the browser above the simulation.

Participants were exposed to designs created by other participants at the top of their browser windows and were shown the best distance that that particular morphology was able to achieve. They were free to ignore these designs or use them as guidance in their own designs. Every time a user clicked *GO* or refreshed the web site, they would be exposed to another random sampling of 13 designs stored in the central repository of designs.

7.3.2 Second Stage: Mining Latent Features

The methodology of designing robots by connecting dots with lines was conducive to storing representations of these designs as a set of edges and nodes in a graph adjacency matrix. We stored the designs of all unique robots created by the crowd in the first stage of the experiment. Then for each of these robot designs, we cal-

culated network measures (see [141] for network measure definitions) using its graph representation (see Table 7.1a for a list of measures used).

We then used these measures calculated on each of the crowdsourced robot morphologies as explanatory variables in a symbolic regression model. The best distance that that morphology was able to move was the response variable used to train the model. We used the *Eureqa* [142] symbolic regression package to build the models. The set of functions allowed are listed in Table 7.1b.

Variable	**Symbol**
Maximal matching	M_{max}
Number of connected components	c
Maximum degree	D_{max}
Minimum degree	D_{min}
Number of limbs	L
Not a chordal graph	C
Symmetry	S
Number of segments	G
Average Degree	D_{ave}
Average degree connectivity	D_{con}
Average clustering	T_{ave}

(a) Variables

Function Name	**Symbol**
Addition	$+$
Subtraction	$-$
Multiplication	$\cdot$
Division	$/$
Logistic function	$\sigma()$
Indicator function	$I()$
Cosine	$cos()$
Sine	$sin()$
Tangent	$tan()$
Exponential	$exp()$
Natural Logarithm	$log()$
Power	x^y
Square root	$\sqrt{}$
Gaussian	$G()$
Less than or equal	$\leq$
Greater than or equal	$\geq$

(b) Functions

Table 7.1: (a) Variables used as explanatory variables and (b) functions allowed for use in symbolic regression expressions.

Eureqa is a multiobjective search tool: it maintains non-dominated solutions along a Pareto front with respect to either minimum error or minimal solution complexity [66]. We ran ten trials of symbolic regression until they reached 100% convergence and

at least 80% *maturity* (proportion of time since the last improvement to a solution, as defined in the *Eureqa* tool) to find expressions that related distance to the set of explanatory network measures. We then selected the best solution of the ten trials with the minimum fit error value (and thus maximal complexity) for use in the third stage of the study.

7.3.3 Third Stage: Seeding the Objective

In the third stage of our study, we used the best expression found using symbolic regression as an additional objective in a genetic algorithm to evolve new robots and their controllers. The genomes in this genetic algorithm consisted of bitstrings grouped into sets of three. Each of these groups corresponded to a potential lines position between adjacent dots in the same 5×5 grid that users in the crowdsourced portion of the study were given to design robots, resulting in a bitstring that consisted of $2 \times 5 \times 4 \times 3 = 120$ bits. Within each group of three bits, the center bit determined whether there was a segment present at that location. If a segment was present (a 1 in the center bit location) the bits to the left and right of the center bit determined whether the motor at the left/top or right/bottom hinge joint would move in-phase or out-of-phase with other motors depending on whether the line was vertical or horizontal (as illustrated in Figure 7.2).

We compared the best distances achieved by two different, *seeded* evolutionary algorithms: both had the primary objective of maximizing the distance that a robot was able to move in a physics simulation, but they were given additional objectives derived from the preferences of the crowdsourced portion of the study. The two treatments differed in this second, seeded, objective. In the first (control) treatment, the

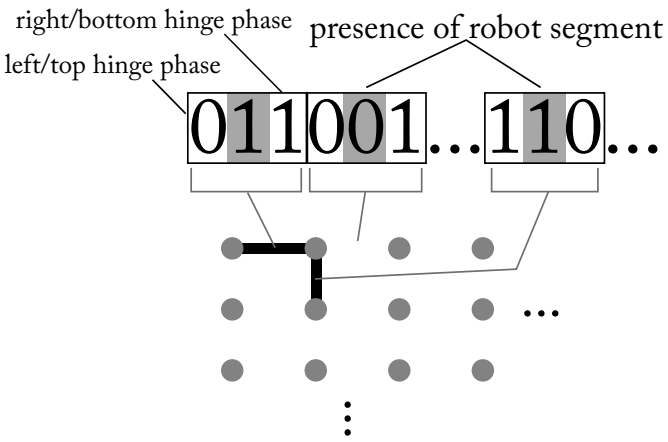

Figure 7.2: ***Genotype to phenotype translation.***

secondary objective was to maximize the ratio of symmetry to number of connected components that made up the robot body. This seeded objective was shown [98] to outperform an evolutionary algorithm with the single objective of maximizing distance that a robot could move. The second (experimental) treatment consisted of using the single best expression found by symbolic regression described in Section 7.3.2 to minimize the error between the expression and distance that a particular design with those parameters could achieve.

We performed 100 independent trials for each of the control and experimental treatments. Each evolutionary process ran for 100 generations with a population of 50 individuals. We used bit-flip mutation at a rate of 0.1 as well as uniform crossover at a rate of 0.1.

7.4 Results

In the first, crowdsourced, stage of the experiment, a total of 947 volunteers participated in robot design. They created 2292 unique designs. On average, 5.63 designs were created by each participant with a long-tailed distribution pattern of participation as is common in crowdsourced studies [60, 143]. Drawings of the best five designs can be seen in Figure 7.3.

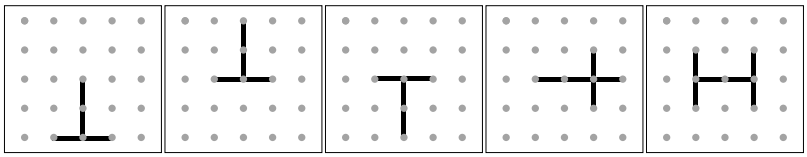

Figure 7.3: ***Top five unique designs in stage one (left to right, top to bottom).*** *Note that some designs that are considered unique are morphologically similar but at different grid coordinates.*

Examples of the ten best symbolic regression expressions found in the second stage of the experiment can be seen in Table 7.2. The expression with the minimum error value (marked with *) was used in the obtaining distance values for the experimental treatment reported in Figure 7.4.

The best distances achieved in the 100 independent runs of the control and experimental treatments in the third stage of the experiment are compared in Figure 7.4. The best five designs in this third stage of the study are illustrated in Figure 7.5.

7.5 Discussion

The introduction of an additional objective using a symbolic regression expression significantly outperformed the inclusion of manually-derived additional objectives to

	Solution	Error
1*	$I(D_{max} \leq L) \cdot min(G(D_{conn}), S) + 0.177min(c + 6.800D_{conn}, D_{max}^{4.510-max(S,log(L))})$	0.771
2	$0.0.041 \cdot c \cdot S \cdot I(L \geq 0.044 \cdot c \cdot D_{max}) + min(c^{0.350}, D_{max})$	0.773
3	$min(D_{max} \cdot \sigma(M_{max}), 2.187) - 0.085 \cdot M_{max}$	0.804
4	$min(3.150, D_{max} + 0.170 \cdot M_{max}) - 0.144 \cdot M_{max}$	0.806
5	$\sqrt{D_{max}} - (0.003 \cdot N \cdot c^2)^S \cdot cos(D_{max})$	0.786
6	$min(3.054, D_{max} \cdot min(M_{max}, 1.325)) - 0.125 \cdot M_{max}$	0.807
7	$4.374 \cdot c \cdot min(0.033, D_{con}) + I(D_{max} \geq 3) + min(S, I(D_{max} \leq L))$	0.781
8	$0.053 \cdot c \cdot S \cdot I(L \geq 0.044 \cdot c \cdot D_{max}) + min(\log c + S, D_{max})$	0.773
9	$D_{con} + S \cdot I(L \geq D_{max}) + 0.196 \cdot min(3.471^{D_{max}}, c) - I(L \geq G)$	0.771
10	$2.819 \cdot \sigma(S) \cdot min(3.180, D_{max}) - 0.141 \cdot M_{max} \cdot min(M_{max}, D_{max}) - 2.960 \cdot I(3.887 - L \geq M_{max})$	0.779

Table 7.2: Lowest error solutions found in the ten symbolic regression trials. See Table 7.1 for variable definitions and the list of functions used in the expressions. Constants are rounded to the nearest thousandth for brevity. Expression 1 was used as the second objective in the experimental treatment.

the fitness in the genetic algorithm as reported in [98] ($p < 0.0001$; independent two group t-test), which was itself shown to significantly outperform the use of the primary objective alone to guide search. We are comparing performance of these methods on the basis of the primary, distance, objective alone. Since adding additional objectives decreases selection pressure on the primary objective, this finding is surprising and encouraging for the method that we introduce here.

The error of the symbolic regression expressions are fairly high. This indicates that either the measures used in the expressions or the expressions themselves were not particularly accurate models of distance traveled by the robots designed by the crowd. Despite this fact, we still achieved a significant improvement by using the expressions as a additional objective in our experiment. This suggests that we might be able to achieve even better results if we were to explore other objective measures of the robot bodies that were discovered by the crowd or if we were to improve the

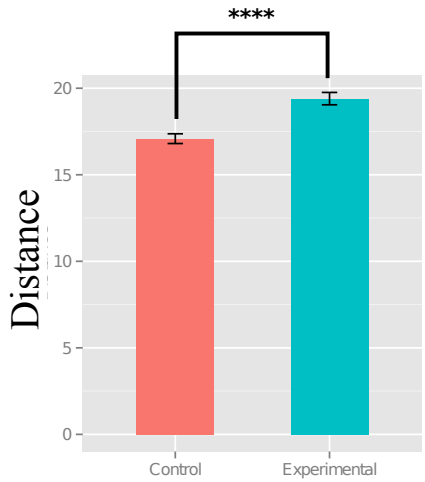

Figure 7.4: ***Distance comparison between treatments.*** *The control case (pink) shows the distance achieved by using the primary distance objective as well as a combined symmetry/number of components objective in* 100 *independent runs of a genetic algorithm. The experimental case (blue) shows the distance achieved by using the primary distance objective as well as the expression obtained using symbolic regression in* 100 *independent runs of the genetic algorithm described in Section 7.3.3. Error bars indicate the 95% confidence intervals around the mean distance value. The difference is significant at the* $p < 0.0001$ *level (independent two group t-test).*

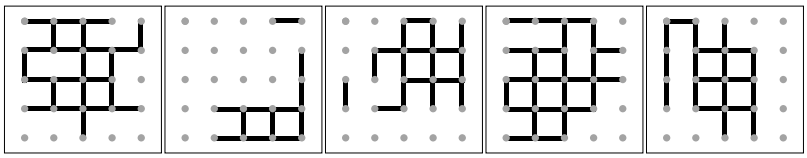

Figure 7.5: ***Top five unique designs in stage three (left to right, top to bottom).***

expressions found by symbolic regression by running evolution longer or fine-tuning genetic programming parameters.

The expressions obtained using symbolic regression are quite complex. We took the expressions from the Pareto Front that had the lowest error of all solutions on the front, which necessarily means that these solutions would be the most complex expressions. Despite their complexity, we have seen empirically that they were able

to outperform more simple multiobjective fitness values. This suggests that there is some valuable latent information in the symbolic regression expressions distilled from the crowd despite the difficulty of parsing out exactly what the expressions entail at an intuitive level.

7.6 Conclusion and Future Work

In this work, we demonstrated a new process for automating the distillation of a crowd's design preferences into an additional objective used to seed an evolutionary algorithm. This objective was in addition to the primary objective of maximizing the distance that a robot was able to move from an initial fixed point in a simulation. We showed that this process significantly outperformed a manual method for extraction of information from crowdsourced studies, which itself was shown to outperform the use of the primary objective alone in the evolutionary algorithm.

In future work, we will investigate whether this technique can be incorporated into other domains. In an era of large amounts of data – much of it generated by humans – there is potential for mining features and expressions from that data that can then be used to improve training of machine learning algorithms, in the development of better design requirements for engineering projects or in seeding algorithms for creative algorithmic work.

In the present study, past designs that were presented to participants consisted of a random sampling of 13 historical designs by other users. In future work, we will investigate whether systematic methods of choosing the historical designs that are shown to participants has an impact on the overall design decisions of the crowd.

Additionally, we will investigate methods to obtain intuitive information from the symbolic regression expressions themselves. In the present study, expressions were directly copied into a multiobjective search. However, there may be kernels of information in the expressions that could be distilled out to further to reduce complexity and drill down to the terms within the expression that are the basis for this method's ability to outperform manually-derived objectives.

Chapter 8

Conclusion

In this thesis we have presented a new framework for distributed human computing. It is different from other distributed human computing systems in that it enables human participants to solve problems by testing, contributing and vetting hypotheses through and in tandem with machine counterparts. In this way, we have enabled machine science through distributed human computing: participants are given means to communicate via the Web, test intuitions via visualizations and web-embedded physics simulations and their efforts are augmented by optimization and machine learning algorithms. This enables them to critically test and share hypotheses with other participants. The ability to formulate, test and communicate hypotheses for critical feedback by other participants parallels the scientific method as proposed in Karl Popper's theory of Critical Rationalism [144]. We rely on the creative capacity and intuition of human participants in tandem with the rapid calculation and communication capabilities of machines to enable complex problem solving by a group of loosely connected individuals over the Web.

The remainder of this chapter summarizes the findings described in detail in pre-

vious chapters and argues the core contribution of the thesis.

Critical Rationalism and the Scientific Method

The origins of scientific hypotheses have long interested scientists themselves as well as philosophers of science [145]. One of the most well-known attempts to describe and formally define where hypotheses come from was offered by Popper. In his treatise on critical rationalism, the philosopher of science, Karl Popper, introduced the following *Tetradic Schema* [144]:

$$P_1 \rightarrow TT \rightarrow EE \rightarrow P_2$$

This schema is a formal description of what Popper believed lie at the heart of the scientific method. It starts with an initial *problem statement* (P_1). From this initial problem statement, a *tentative theory* (TT) is formulated, commonly referred to as a hypothesis. The tentative theory is an approach to or idea for a solution to the problem at hand. After the tentative theory has been proposed, an iterative process of *error elimination* (EE) is conducted in order to arrive at another problem formulation (P_2), which exists at a more advanced level of knowledge than the initial problem statement, P_1. This procedure for achieving an advanced state of knowledge or problem formulation ability (P_2) from the current statement of a problem (P_1) is through an iterative process of hypothesis formulation (TT) and testing (EE). Iterative critique of a stated hypothesis must take place at the subjective level, but

social critique is equally important. In the subjective critique of a hypothesis, we expose our own theory to the possibility that it does not hold. But requiring a hypothesis to stand up to social critique is also necessary. In this way, we can subject our hypotheses to error elimination from alternate perspectives.

At the core of the methodology we have introduced in this thesis is an algorithm that invokes Popper's Tetradic Schema as inspiration. We enable collective efforts for hypothesis discovery by: presenting participants with a problem; enabling them to test their hypotheses through a web-based tool; encouraging the dissemination of their hypothesis for social critique and to act as inspiration or dissuasion for other participants. As participants suggest hypotheses and obtain results from tests, new hypotheses can be based on previous user-suggested hypotheses.

We used varying mechanisms through which a crowd of non-expert contributors could suggest hypotheses. In the domain of robot control design, we asked participants to connect joints on a diagram. All joints connected by these lines would move together in a physics simulation. In this way, a user of our tool is a hypothesis for what that user believed might lead to rapid forward locomotion in a robot. The hypothesis was a graphical one, thus enabling the hypothesis to be communicated to other participants for social vetting. Utilizing the same graphical means for hypothesis suggestion, we asked users to suggest hypotheses for what robot morphology could lead to rapid forward locomotion. We also asked users to suggest questions as hypotheses for what behaviors drive residential energy usage. In this way, citizen scientists were able to participate in hypothesis generation and vetting to develop a predictive model by simply asking questions. We could thus ask non-scientist participants to engage in the scientific method via the Web.

We have enabled a group of individuals to formulate, test and vet hypotheses via the Web. And we demonstrated that they were able to better formulate hypotheses with the aid of machines. They could communicate over the Web, they worked in tandem with machine learning algorithms, and their hypotheses were combined to be used as an objective in an automated robot design algorithm and predictive model. Thus in tandem with machine capabilities, such as rapid communication over the Web and machine learning algorithms, we have shown that a group of loosely-connected, anonymous individuals are given enhanced means through which they may generate hypotheses.

Machine Science

At the basis of the methodology introduced in the previous chapters is enabling the crowd to generate, test, communicate and vet hypotheses over the Web and with the aid of machines. This methodology falls under the definition of 'machine science' in which human hypothesis generation is augmented through the use of machines.

Swanson suggests [35, 36] that there exists knowledge that is yet undiscovered in the public sphere but is available to be mined should we know to look for it. This public knowledge exists in the logical conclusions that can be formulated through the combination of existing knowledge. In objective knowledge we find publicly available hypotheses to be recombined into new knowledge or hypotheses.

It is in this spirit that the definition of machine science was formulated. Machine science refers to the use of machine proficiencies to enhance the capability for scientific discovery and better solve problems [34]. In any such system, humans serve the role

of creative, hypothesis-contributing agents.

Distributed Human Computing Methods and Machine Science

Throughout this thesis we have used distributed human computing techniques to enable hypothesis generation, testing, communication and vetting by a group of anonymous Web users. Here we discuss the connection between distributed human computing and machine science.

It has already been demonstrated that distributed human computing methods such as crowdsourcing can be effective at distributing simple tasks to a large group of non-expert individuals anonymously over the web. By demonstrating that machine science is possible through distributed human computing, we show that these methods can indeed scale up to more complex tasks. Thus we have demonstrated one methodology that can be used to move distributed human computing methods from a means for completing simple tasks to one of enabling the crowd to solve more difficult problems. We did this by demonstrating the effectiveness of a method through which distributed human computation can be used to solve problems previously deemed only possible for a team of scientists. This collaborative methodology is a blackboard system-like framework [49, 146] for human agents to contribute hypotheses to solving a problem. Users of a web-based system are given both the capability to test their own hypothesis and contribute a hypothesis for others to test. In the case of robot control and morphology design, a panel of historical designs was displayed for use by others in their group along the top of the web-embedded design tool. Thus users were

free to use, improve upon or ignore those hypotheses contributed by other participants. Thus, rather than the traditional hub and spoke crowdsourcing architecture employed by most distributed human computing systems, we introduced a way for the crowd to communicate among themselves. And, as such, be able to generate and advertise hypotheses, as well as test those hypotheses as much as they wished using their local computer. The computer itself determines whether the test results in support for, or evidence against, the given hypothesis. We formally demonstrated that this methodology outperforms a crowdsourcing mechanism in which the task of hypothesis formulation is distributed to individual participants.

However, we have only demonstrated that this method can be useful under specific circumstances. Namely, we have demonstrated that collectives can productively contribute and vet hypotheses through social, online venues when they have some basic degree of familiarity or intuition with the problem presented to them. This need for a minimal level of sophistication of the crowd with respect to the problem presented to them has been referred to as the *Calculus Condition* [37]. In this work, we specifically chose the problem of building robots through the Web because the crowd has intuition for locomotion. Being embodied in Nature, humans have experience with animal locomotion. We hypothesized that this would in turn inform the hypotheses that they contributed. The ability to formulate hypotheses about home energy usage was also considered one with which the crowd has some degree of familiarity. This is because they were all account holders on residential electricity bills and very likely had experience with cost of energy usage and behavior. Demonstrating the conditions under which a collaborative crowdsourcing system does not result in a constructive social feedback process of hypothesis formulation has yet to be investigated.

In this work we have also contributed a methodology that we termed *crowdseeding*. Crowdseeding is a process for incorporating crowd feedback into a machine learning or optimization algorithm. By utilizing crowd tendencies as an additional optimization objective in a robot design task, we demonstrated that crowd contributions can result in improved performance of robot locomotion. In fact, in Chapter 6, we showed that the crowd collectively favored an objective that has been demonstrated in past scientific literature to aid in discovery of optimal robot gaits [101], unbeknownst to the crowd. Several users independently favored symmetric designs, thus reducing the likelihood that one lone contributor proposed a symmetric design that then infected the designs of everyone else. However, again, we emphasize that we have demonstrated a specific example for how crowdseeding can be an effective method for incorporating crowd feedback into an optimization algorithm. But we have not necessarily shown that crowd feedback always results in a helpful additional objective in an optimization task.

Implications of this work

In this thesis, we have demonstrated specific applications for how machine science can be enabled through distributed human computing. These are applications that exploit common intuition among users of the Web: experience with animal locomotion and familiarity with electric energy usage. Despite the specific applications, the general framework in which hypotheses are generated, tested, communicated and vetted has historically been a useful means for problem solving and discovery. And demonstrating that the use of machines to aid this process in the examples within

this thesis suggests potential for other application domains.

Both existing and new applications could benefit from a system described in this thesis. The protein folding game, FoldIt [14], asked users to fold proteins. In this way, they developed recipes to reconfigure proteins for medical applications. However, the only interaction between players was through a scoreboard and via recipes in a separate tool whose quality was not explicitly communicated for vetting. Enabling players to succinctly communicate graphical hypotheses for configuration strategies for vetting in the way presented here might have a beneficial impact on players.

Asking the crowd to test hypotheses requires a forward model for evaluation. Thus, some objective measure for evaluating hypotheses is necessary for any application of the strategy presented in this thesis. Thus this methodology is appropriate for domains that can be modeled by a machine, even if only at a simplified level. For example, political processes, such as those that are amenable to game theoretic or agent-based models could be used as the evaluation criteria for crowd-suggested hypotheses for solving political or behavioral problems.

We utilized the crowdseeding technique for the purpose of designing robots. We showed that symmetry was one component of crowd preferences that could be used to design robots that were better able to locomote than if we were to utilize only distance as an objective for robot design. We might presume that the crowd favored symmetry due to their experience with walking animals in Nature. Other domains may or may not benefit from such natural crowd biases. Therefore, relying on automated means by which crowd biases can be distilled and then exposed for testing and vetting will be important across domain applications. Crowdseeding could be used as a component in a feedback loop that distills information from the crowd, and then

presents that information to the crowd. This feedback mechanism might help to illuminate overall crowd biases. Finding the optimal trade-off between exploiting these biases and exploring ways of overcoming them may lead to new recursive forms of the crowdseeding methodology.

Summary

The core contributions of this work were to demonstrate that distributed human computing or collective intelligence methods can be used to enable machine science. A methodology was introduced for how this is possible, which is loosely based on Popper's Tetradic Schema: a problem is presented; individuals are enabled through Web-based means to create and test hypotheses for how best to solve the problem; and hypotheses are communicated to other participants for critique and can trigger the formulation of new hypotheses by other members of the group. We demonstrated that this is possible through the design of robot control strategies by a group of anonymous participants over the Web. We also showed that the crowd was capable of collectively designing robot bodies over the Web that outperformed current state-of-the-art algorithms. We demonstrated that this methodology indeed benefited from social feedback: isolated individuals were less capable than if they had the benefit of social participation. And we showed that machine science over the Web is not limited to robotics: the crowd contributed useful predictive features to a model of residential energy consumption.

In addition to this methodology for enabling machine science directly through collective intelligence methods over the Web, *crowdseeding* was introduced. Crowdseed-

ing refers to the process of incorporating crowd preferences as an additional objective in a machine learning or optimization algorithm. The method was demonstrated to improve the ability of an optimization algorithm to design of robot morphologies by distilling and utilizing crowd preferences.

It is clear that the amount, diversity, frequency and intimacy of biological and computational interaction will continue to grow in the years and decades to come. As just one example, direct brain-to-computer interaction is already a reality [147, 148]. The work presented here provides one way of thinking about how to establish such interactions so that synergies are maximized and antagonisms are minimized.

Appendix A

Appendix

Symmetry Measure Derivation

Completely symmetric designs in the space of robot morphologies possible with our characterization are few in number. The total number of robot designs possible on a 5-by-5 grid are $2^{(5\cdot4+5\cdot4)} = 2^{40} \approx 1.1 \times 10^{12}$. The number of bounding boxes (the smallest rectangular box that can enclose a given design) of d_1 grid units by d_2 grid units in a $D \times D$ grid is $(D-d_1+1)(D-d_2+1)$. Of each of these bounding boxes, we can have either vertical, horizontal or either of two diagonal lines of symmetry. The diagonal lines of symmetry were only considered if $d_1 = d_2$. The number of perfect horizontal symmetries are the number of designs that fit into one half of the bounding box that can be reflected about the vertical line of symmetry:

$$H_{sym} = \begin{cases} 2^{\lfloor \frac{d_1}{2} \rfloor (2d_2-1)} & : \quad d_1 \text{ odd} \\ 2^{d_1(d_2-1)} & : \quad d_1 \text{ even} \end{cases}$$

Similarly for vertical symmetry in the bounding box:

$$V_{sym} = \begin{cases} 2^{\lfloor \frac{d_2}{2} \rfloor (2d_1 - 1)} & : \quad d_2 \text{ odd} \\ 2^{d_2(d_1-1)} & : \quad d_2 \text{ even} \end{cases}$$

The number of symmetries for both diagonal axes when the bounding box is square is

$$D_{sym} = 2^{d(d-1)}, \text{ where } d = d_1 = d_2.$$

The total number of perfectly symmetric designs in a grid of size D are thus:

$$\text{Total \# symmetries} = \sum_{d_1=1}^{D} \sum_{d_2=1}^{D} (D - d_1 + 1)(D - d_2 + 1)(H_{sym} + V_{sym} + D_{sym})$$

For a 5-by-5 grid, the number of perfect symmetries (defined by reflections about an axis of symmetry in bounding boxes) is approximately 1.97 million. This amounts to ~ 1.82 perfectly symmetric designs per million designs in this space of all possible robot designs.

Bibliography

[1] Bei Yu, Matt Willis, Peiyuan Sun, and Jun Wang. Crowdsourcing participatory evaluation of medical pictograms using amazon mechanical turk. *Journal of medical Internet research*, 15(6):e108, 2013.

[2] Joseph K Goodman, Cynthia E Cryder, and Amar Cheema. Data collection in a flat world: The strengths and weaknesses of mechanical turk samples. *Journal of Behavioral Decision Making*, 26(3):213–224, 2013.

[3] Iain Couzin. Collective minds. *Nature*, 445(7129):715–715, 2007.

[4] Rolf Pfeifer and Josh Bongard. *How the body shapes the way we think: a new view of intelligence*. MIT press, 2006.

[5] Alexander J Quinn and Benjamin B Bederson. A taxonomy of distributed human computation. *Human-Computer Interaction Lab Tech Report, University of Maryland*, 2009.

[6] Alexander J Quinn and Benjamin B Bederson. Human computation: a survey and taxonomy of a growing field. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1403–1412. ACM, 2011.

[7] Jeff Howe. The rise of crowdsourcing. *Wired magazine*, 14(6):1–4, 2006.

[8] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the conference on empirical methods in natural language processing*, pages 254–263. Association for Computational Linguistics, 2008.

[9] Luis Von Ahn. Human computation. In *Design Automation Conference, 2009. DAC'09. 46th ACM/IEEE*, pages 418–419. IEEE, 2009.

[10] Luis Von Ahn, Benjamin Maurer, Colin McMillen, David Abraham, and Manuel Blum. Reaptcha: Human-based character recognition via web security measures. *Science*, 321(5895):1465–1468, 2008.

[11] Luis Von Ahn and Laura Dabbish. Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 319–326. ACM, 2004.

[12] Doug Schuler. Social computing. *Communications of the ACM*, 37(1):28–29, 1994.

[13] Fei-Yue Wang, Kathleen M Carley, Daniel Zeng, and Wenji Mao. Social computing: From social informatics to social intelligence. *Intelligent Systems, IEEE*, 22(2):79–83, 2007.

[14] Firas Khatib, Seth Cooper, Michael D Tyka, Kefan Xu, Ilya Makedon, Zoran Popović, David Baker, and Foldit Players. Algorithm discovery by protein folding game players. *Proceedings of the National Academy of Sciences*, 108(47):18949–18953, 2011.

[15] Chris J Lintott, Kevin Schawinski, Anže Slosar, Kate Land, Steven Bamford, Daniel Thomas, M Jordan Raddick, Robert C Nichol, Alex Szalay, Dan Andreescu, et al. Galaxy zoo: morphologies derived from visual inspection of galaxies from the sloan digital sky survey. *Monthly Notices of the Royal Astronomical Society*, 389(3):1179–1189, 2008.

[16] Jinseop S Kim, Matthew J Greene, Aleksandar Zlateski, Kisuk Lee, Mark Richardson, Srinivas C Turaga, Michael Purcaro, Matthew Balkam, Amy Robinson, Bardia F Behabadi, et al. Space-time wiring specificity supports direction selectivity in the retina. *Nature*, 509(7500):331–336, 2014.

[17] Luis Von Ahn. Games with a purpose. *Computer*, 39(6):92–94, 2006.

[18] Morton E O'Kelly. A geographer's analysis of hub-and-spoke networks. *Journal of transport Geography*, 6(3):171–186, 1998.

[19] Wolfgang Banzhaf. Interactive evolution. *Evolutionary Computation*, 1:228–236, 2000.

[20] Artemis Moroni, Jônatas Manzolli, Fernando Von Zuben, and Ricardo Gudwin. Vox populi: An interactive evolutionary system for algorithmic music composition. *Leonardo Music Journal*, 10:49–54, 2000.

[21] Nao Tokui and Hitoshi Iba. Music composition with interactive evolutionary computation. In *Proceedings of the 3rd international conference on generative art*, volume 17, pages 215–226, 2000.

[22] Henrik Hautop Lund, Orazio Miglino, Luigi Pagliarini, Aude Billard, and Auke Ijspeert. Evolutionary robotics-a children's game. In *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, pages 154–158. IEEE, 1998.

[23] Henrik Hautop Lund and Orazio Miglino. Evolving and breeding robots. In *Evolutionary Robotics*, pages 192–210. Springer, 1998.

[24] Sonia Chernova, Nick DePalma, Elisabeth Morant, and Cynthia Breazeal. Crowdsourcing human-robot interaction: Application from virtual to physical worlds. In *RO-MAN, 2011 IEEE*, pages 21–26. IEEE, 2011.

[25] Richard Dawkins. *The blind watchmaker: Why the evidence of evolution reveals a universe without design*. WW Norton & Company, 1996.

[26] Hideyuki Takagi. Interactive evolutionary computation: System optimization based on human subjective evaluation. In *IEEE Int. Conf. on Intelligent Engineering Systems*, pages 17–19, 1998.

[27] Alex Kosorukoff. Human based genetic algorithm. In *Systems, Man, and Cybernetics, 2001 IEEE International Conference on*, volume 5, pages 3464–3469. IEEE, 2001.

[28] Boyang Li, Darren Scott Appling, Stephen Lee-Urban, and Mark O Riedl. Learning sociocultural knowledge via crowdsourced examples. In *Proc. of the 4th AAAI Workshop on Human Computation*, 2012.

[29] Jeff Orkin and Deb Roy. Automatic learning and generation of social behavior from collective human gameplay. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 385–392. International Foundation for Autonomous Agents and Multiagent Systems, 2009.

[30] Push Singh and William Williams. Lifenet: a propositional model of ordinary human activity. In *Proceedings of the Workshop on Distributed and Collaborative Knowledge Capture (DC-KCAP) at KCAP*, volume 2003, 2003.

[31] Push Singh, Thomas Lin, Erik T Mueller, Grace Lim, Travell Perkins, and Wan Li Zhu. Open mind common sense: Knowledge acquisition from the general public. In *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*, pages 1223–1237. Springer, 2002.

[32] Jim Giles. Internet encyclopaedias go head to head. *Nature*, 438(7070):900–901, 2005.

[33] Timothy Gowers and Michael Nielsen. Massively collaborative mathematics. *Nature*, 461(7266):879–881, 2009.

[34] James Evans and Andrey Rzhetsky. Philosophy of science: Machine science. *Science (New York, NY)*, 329(5990):399, 2010.

[35] Don R Swanson. Undiscovered public knowledge. *The Library Quarterly*, pages 103–118, 1986.

[36] Don R Swanson. Fish oil, raynaud's syndrome, and undiscovered public knowledge. *Perspectives in biology and medicine*, 30(1):7–18, 1986.

[37] Scott E Page. *The difference: How the power of diversity creates better groups, firms, schools, and societies*. Princeton University Press, 2008.

[38] Irving L Janis. *Victims of groupthink: A psychological study of foreign-policy decisions and fiascoes.* Houghton Mifflin, 1972.

[39] Alan G Ingham, George Levinger, James Graves, and Vaughn Peckham. The ringelmann effect: Studies of group size and group performance. *Journal of Experimental Social Psychology*, 10(4):371–384, 1974.

[40] Anita Williams Woolley, Christopher F Chabris, Alex Pentland, Nada Hashmi, and Thomas W Malone. Evidence for a collective intelligence factor in the performance of human groups. *Science*, 330(6004):686–688, 2010.

[41] Thomas R. G. Green and Marian Petre. Usability analysis of visual programming environments: a cognitive dimensions framework. *Journal of Visual Languages & Computing*, 7(2):131–174, 1996.

[42] Gerry Dozier. Evolving robot behavior via interactive evolutionary computation: From real-world to simulation. In *Proceedings of the 2001 ACM symposium on Applied computing*, pages 340–344. ACM, 2001.

[43] Shane Celis, Gregory S Hornby, and Josh Bongard. Avoiding local optima with user demonstrations and low-level control. In *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pages 3403–3410. IEEE, 2013.

[44] Josh C Bongard and Gregory S Hornby. Combining fitness-based search and user modeling in evolutionary robotics. In *Proceeding of the fifteenth annual conference on Genetic and evolutionary computation conference*, pages 159–166. ACM, 2013.

[45] Riad Akrour, Marc Schoenauer, and Michele Sebag. Preference-based policy learning. In *Machine Learning and Knowledge Discovery in Databases*, pages 12–27. Springer, 2011.

[46] Taras Kowaliw, Alan Dorin, and Jon McCormack. Promoting creative design in interactive evolutionary computation. *IEEE transactions on evolutionary computation*, 16(4):523, 2012.

[47] Mario García-Valdez, Leonardo Trujillo, Francisco Fernández de Vega, Juan Julián Merelo Guervós, and Gustavo Olague. Evospace-interactive: a framework to develop distributed collaborative-interactive evolutionary algorithms for artistic design. In *Evolutionary and Biologically Inspired Music, Sound, Art and Design*, pages 121–132. Springer, 2013.

[48] Jill H Larkin. Display-based problem solving. *Complex information processing: The impact of Herbert A. Simon*, pages 319–341, 1989.

[49] H Penny Nii. The blackboard model of problem solving and the evolution of blackboard architectures. *AI magazine*, 7(2):38, 1986.

[50] John Peterson, Gregory D Hager, and Paul Hudak. A language for declarative robotic programming. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 2, pages 1144–1151. IEEE, 1999.

[51] Philip T Cox and Trevor J Smedley. Visual programming for robot control. In *Visual Languages, 1998. Proceedings. 1998 IEEE Symposium on*, pages 217–224. IEEE, 1998.

[52] Philip T Cox, Christopher C Risley, and Trevor J Smedley. Toward concrete representation in visual languages for robot control. *Journal of Visual Languages & Computing*, 9(2):211–239, 1998.

[53] Joseph J Pfeiffer Jr. Altaira: A rule-based visual language for small mobile robots. *Journal of Visual Languages and Computing*, 9(2):127–150, 1998.

[54] HY Kan, Vincent G Duffy, and Chuan-Jun Su. An internet virtual reality collaborative environment for effective product design. *Computers in Industry*, 45(2):197–213, 2001.

[55] Anhai Doan, Raghu Ramakrishnan, and Alon Y Halevy. Crowdsourcing systems on the world-wide web. *Communications of the ACM*, 54(4):86–96, 2011.

[56] Lixiu Yu and Jeffrey Nickerson. Generating creative ideas through crowds: An experimental study of combination. In *Thirty Second International Conference on Information Systems*, 2011.

[57] Jimmy Secretan, Nicholas Beato, David B D'Ambrosio, Adelein Rodriguez, Adam Campbell, Jeremiah T Folsom-Kovarik, and Kenneth O Stanley. Picbreeder: A case study in collaborative evolutionary exploration of design space. *Evolutionary Computation*, 19(3):373–403, 2011.

[58] Kenneth O Stanley and Risto Miikkulainen. Competitive coevolution through evolutionary complexification. *J. Artif. Intell. Res.(JAIR)*, 21:63–100, 2004.

[59] Sebastian Deterding, Miguel Sicart, Lennart Nacke, Kenton O'Hara, and Dan Dixon. Gamification. using game-design elements in non-gaming contexts. In *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems*, pages 2425–2428. ACM, 2011.

[60] Dennis M Wilkinson. Strong regularities in online peer production. In *Proceedings of the 9th ACM conference on Electronic commerce*, pages 302–309. ACM, 2008.

[61] Raffaello D'Andrea. Guest editorial can drones deliver? *IEEE Transactions on Automation Science and Engineering*, 11(3):647–648, 2014.

[62] Justin Werfel, Kirstin Petersen, and Radhika Nagpal. Designing collective behavior in a termite-inspired robot construction team. *Science*, 343(6172):754–758, 2014.

[63] Yael Edan, Shufeng Han, and Naoshi Kondo. Automation in agriculture. In *Springer Handbook of Automation*, pages 1095–1128. Springer, 2009.

[64] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[65] Samik Ghosh, Yukiko Matsuoka, Yoshiyuki Asai, Kun-Yi Hsin, and Hiroaki Kitano. Software for systems biology: from tools to integrated platforms. *Nature Reviews Genetics*, 12(12):821–832, 2011.

[66] Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, 2009.

[67] Sylvain Gelly, Levente Kocsis, Marc Schoenauer, Michele Sebag, David Silver, Csaba Szepesvári, and Olivier Teytaud. The grand challenge of computer

go: Monte carlo tree search and extensions. *Communications of the ACM*, 55(3):106–113, 2012.

[68] Jimmy Secretan, Nicholas Beato, David B D Ambrosio, Adelein Rodriguez, Adam Campbell, and Kenneth O Stanley. Picbreeder: evolving pictures collaboratively online. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1759–1768. ACM, 2008.

[69] Mark Wagy and Josh Bongard. Collective design of robot locomotion. In *ALIFE 14: The Fourteenth Conference on the Synthesis and Simulation of Living Systems*, volume 14, pages 138–145, 2014.

[70] Brian G Woolley and Kenneth O Stanley. A novel human-computer collaboration: combining novelty search with interactive evolution. In *Proceedings of the 2014 conference on Genetic and evolutionary computation*, pages 233–240. ACM, 2014.

[71] Cynthia Breazeal, Nick DePalma, Jeff Orkin, Sonia Chernova, and Malte Jung. Crowdsourcing human-robot interaction: New methods and system evaluation in a public environment. *Journal of Human-Robot Interaction*, 2(1):82–111, 2013.

[72] Alexander Sorokin, Dmitry Berenson, Siddhartha S Srinivasa, and Martial Hebert. People helping robots helping people: Crowdsourcing for grasping novel objects. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 2117–2122. IEEE, 2010.

[73] George Mather and Sophie West. Recognition of animal locomotion from dynamic point-light displays. *Perception*, 22:759–759, 1993.

[74] Peter Neri, M Concetta Morrone, and David C Burr. Seeing biological motion. *Nature*, 395(6705):894–896, 1998.

[75] Gunnar Johansson. Visual perception of biological motion and a model for its analysis. *Perception & psychophysics*, 14(2):201–211, 1973.

[76] Maeve Duggan and Aaron Smith. 6% of online adults are reddit users. *Pew Internet & American Life Project*, 3, 2013.

[77] Marlene E Turner and Anthony R Pratkanis. Twenty-five years of groupthink theory and research: Lessons from the evaluation of a theory. *Organizational behavior and human decision processes*, 73(2):105–115, 1998.

[78] Michael Klug and James P Bagrow. Understanding the group dynamics and success of teams. *arXiv preprint arXiv:1407.2893*, 2014.

[79] Alex Pentland. The new science of building great teams. *Harvard Business Review*, 90(4):60–69, 2012.

[80] Kenneth O Stanley, David B D'Ambrosio, and Jason Gauci. A hypercube-based encoding for evolving large-scale neural networks. *Artificial life*, 15(2):185–212, 2009.

[81] Joshua E Auerbach and Josh C Bongard. Evolving cppns to grow three-dimensional physical structures. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 627–634. ACM, 2010.

[82] Joshua E Auerbach and Josh C Bongard. Evolving complete robots with cppn-neat: the utility of recurrent connections. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 1475–1482. ACM, 2011.

[83] Nick Cheney, Robert MacCurdy, Jeff Clune, and Hod Lipson. Unshackling evolution: evolving soft robots with multiple materials and a powerful generative encoding. In *Proceeding of the fifteenth annual conference on Genetic and evolutionary computation conference*, pages 167–174. ACM, 2013.

[84] Jeff Clune, Benjamin E Beckmann, Charles Ofria, and Robert T Pennock. Evolving coordinated quadruped gaits with the hyperneat generative encoding. In *IEEE Congress on Evolutionary Computation, 2009. CEC'09.*, pages 2764–2771. IEEE, 2009.

[85] Evert Haasdijk, Andrei A Rusu, and AE Eiben. Hyperneat for locomotion control in modular robots. In *Evolvable systems: from biology to hardware*, pages 169–180. Springer, 2010.

[86] Kenneth O Stanley. Compositional pattern producing networks: A novel abstraction of development. *Genetic programming and evolvable machines*, 8(2):131–162, 2007.

[87] Sebastian Risi, Daniel Cellucci, and Hod Lipson. Ribosomal robots: Evolved designs inspired by protein folding. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, pages 263–270. ACM, 2013.

[88] John R Finnerty. Did internal transport, rather than directed locomotion, favor the evolution of bilateral symmetry in animals? *BioEssays*, 27(11):1174–1180, 2005.

[89] Lashon B. Booker, David E. Goldberg, and John H. Holland. Classifier systems and genetic algorithms. *Artificial intelligence*, 40(1):235–282, 1989.

[90] Hod Lipson and Melba Kurman. *Fabricated: The new world of 3D printing*. John Wiley & Sons, 2013.

[91] Joshua G Tanenbaum, Amanda M Williams, Audrey Desjardins, and Karen Tanenbaum. Democratizing technology: pleasure, utility and expressiveness in diy and maker practice. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2603–2612. ACM, 2013.

[92] Paul DiMaggio, Eszter Hargittai, W Russell Neuman, and John P Robinson. Social implications of the internet. *Annual review of sociology*, pages 307–336, 2001.

[93] Jared Moore, Anthony Clark, and Philip McKinley. Evolutionary robotics on the web with webgl and javascript. *arXiv preprint arXiv:1406.3337*, 2014.

[94] Manoj Parameswaran and Andrew B Whinston. Social computing: An overview. *Communications of the Association for Information Systems*, 19(1):37, 2007.

[95] Robin IM Dunbar. Neocortex size as a constraint on group size in primates. *Journal of Human Evolution*, 22(6):469–493, 1992.

[96] Bruno Gonçalves, Nicola Perra, and Alessandro Vespignani. Modeling users' activity on twitter networks: Validation of dunbar's number. *PloS one*, 6(8):e22656, 2011.

[97] Mark D Wagy and Josh C Bongard. Combining computational and social effort for collaborative problem solving. *PloS one*, 10(11):e0142524, 2015.

[98] Mark D Wagy and Josh Bongard. Crowdseeding robot design. In *Proceedings of the 24th annual conference on Genetic and evolutionary computation*. ACM, forthcoming.

[99] Mark D Wagy and Josh C Bongard. Crowdseeding: a novel approach for designing bioinspired machines. In *Biomimetic and Biohybrid Systems*, pages 293–303. Springer, 2015.

[100] Anand Rajaraman, Jeffrey D Ullman, Jeffrey David Ullman, and Jeffrey David Ullman. *Mining of massive datasets*, volume 1. Cambridge University Press Cambridge, 2012.

[101] Josh C Bongard and Chandana Paul. Investigating morphological symmetry and locomotive efficiency using virtual embodied evolution. In *From Animals to Animats: The Sixth International Conference on the Simulation of Adaptive Behaviour*. Citeseer, 2000.

[102] Robert Swain, Alex Berger, Josh Bongard, and Paul Hines. Participation and contribution in crowdsourced surveys. *PloS one*, 10(4):e0120521, 2015.

[103] Stamatis Karnouskos, Orestis Terzidis, and Panagiotis Karnouskos. An advanced metering infrastructure for future energy networks. In *New Technologies, Mobility and Security*, pages 597–606. Springer, 2007.

[104] Ian Ayres, Sophie Raseman, and Alice Shih. Evidence from two large field experiments that peer comparison feedback can reduce residential energy usage. *Journal of Law, Economics, and Organization*, 2012.

[105] Ramyar Rashed Mohassel, Alan Fung, Farah Mohammadi, and Kaamran Raahemifar. A survey on advanced metering infrastructure. *International Journal of Electrical Power & Energy Systems*, 63:473–484, 2014.

[106] Shahzeen Z. Attari, Michael L. DeKay, Cliff I. Davidson, and Wändi Bruine de Bruinc. Public perceptions of energy consumption and savings. *Proceedings of the National Academy of Sciences*, 107(37):16054–16059, 2010.

[107] Omar I. Asensio and Magali A. Delmas. Nonprice incentives and energy conservation. *Proceedings of the National Academy of Sciences*, 112(6):E510–E515, 2015.

[108] Thomas F. Sanquist, Heather Orr, Bin Shui, and Alvah C. Bittner. Lifestyle factors in u.s. residential electricity consumption. *Energy Policy*, 42:354 – 364, 2012.

[109] Christopher A. Craig and Myria W. Allen. Enhanced understanding of energy ratepayers: Factors influencing perceptions of government energy efficiency subsidies and utility alternative energy use. *Energy Policy*, 66(0):224 – 233, 2014.

[110] Enrico Costanza, Sarvapali D Ramchurn, and Nicholas R Jennings. Understanding domestic energy consumption through interactive visualisation: a field study. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 216–225. ACM, 2012.

[111] Tri Kurniawan Wijaya, Matteo Vasirani, and Karl Aberer. Crowdsourcing behavioral incentives for pervasive demand response. Technical report, 2014.

[112] Josh C Bongard, Paul DH Hines, Dylan Conger, Peter Hurd, and Zhenyu Lu. Crowdsourcing predictors of behavioral outcomes. *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, 43(1):176–185, 2013.

[113] Kirsten E Bevelander, Kirsikka Kaipainen, Robert Swain, Simone Dohle, Josh C Bongard, PD Hines, and Brian Wansink. Crowdsourcing novel childhood predictors of adult obesity. *PloS one*, 9(2):e87756, 2014.

[114] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[115] Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002.

[116] Thomas G Dietterich. Ensemble methods in machine learning. In *Multiple classifier systems*, pages 1–15. Springer, 2000.

[117] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[118] Roderick JA Little and Donald B Rubin. *Statistical Analysis with Missing Data*. John Wiley & Sons, 2014.

[119] Kellie J Archer and Ryan V Kimes. Empirical characterization of random forest variable importance measures. *Computational Statistics & Data Analysis*, 52(4):2249–2260, 2008.

[120] Peter Hall and Michael G Schimek. Moderate-deviation-based inference for random degeneration in paired rank lists. *Journal of the American Statistical Association*, 107(498):661–672, 2012.

[121] Michael G Schimek, Eva Budinská, Karl G Kugler, Vendula Švendová, Jie Ding, and Shili Lin. Topklists: a comprehensive r package for statistical inference, stochastic aggregation, and visualization of multiple omics ranked lists. *Statistical Applications in Genetics and Molecular Biology*, 14(3):311–316, 2015.

[122] James Surowiecki, Mark P Silverman, et al. The wisdom of crowds. *American Journal of Physics*, 75(2):190–192, 2007.

[123] Aniket Kittur. Crowdsourcing, collaboration and creativity. *ACM Crossroads*, 17(2):22–26, 2010.

[124] Huiji Gao, Geoffrey Barbier, Rebecca Goolsby, and Daniel Zeng. Harnessing the crowdsourcing power of social media for disaster relief. Technical report, DTIC Document, 2011.

[125] Luis Von Ahn, Ruoran Liu, and Manuel Blum. Peekaboom: a game for locating objects in images. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 55–64. ACM, 2006.

[126] Ece Kamar, Severin Hacker, and Eric Horvitz. Combining human and machine intelligence in large-scale crowdsourcing. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 467–474. International Foundation for Autonomous Agents and Multiagent Systems, 2012.

[127] Hideyuki Takagi. Interactive evolutionary computation: Fusion of the capabilities of ec optimization and human evaluation. *Proceedings of the IEEE*, 89(9):1275–1296, 2001.

[128] Kenneth O Stanley. Exploiting regularity without development. In *Proceedings of the AAAI Fall Symposium on Developmental Systems*, page 37. AAAI Press Menlo Park, CA, 2006.

[129] David Sanders and Yoshihiro Kusuda. Toyota's violin-playing robot. *Industrial Robot: An International Journal*, 35(6):504–506, 2008.

[130] Kazuo Hirai, Masato Hirose, Yuji Haikawa, and Toru Takenaka. The development of honda humanoid robot. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 2, pages 1321–1326. IEEE, 1998.

[131] Kenji Kaneko, Kensuke Harada, Fumio Kanehiro, Gou Miyamori, and Kazuhiko Akachi. Humanoid robot hrp-3. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 2471–2478. IEEE, 2008.

[132] Sebastian Lohmeier, Thomas Buschmann, and Heinz Ulbrich. Humanoid robot lola. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 775–780. IEEE, 2009.

[133] Marc Raibert, Kevin Blankespoor, Gabriel Nelson, Rob Playter, et al. Bigdog, the rough-terrain quadruped robot. In *Proceedings of the 17th World Congress*, volume 17, pages 10822–10825, 2008.

[134] Johann Borenstein, Malik Hansen, and Adam Borrell. The omnitread ot-4 serpentine robot design and performance. *Journal of Field Robotics*, 24(7):601–621, 2007.

[135] Gregory Dudek, Michael Jenkin, Chris Prahacs, Andrew Hogue, Junaed Sattar, Philippe Giguere, Andrew German, Hui Liu, Shane Saunderson, Arlene Ripsman, et al. A visually guided swimming robot. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 3604–3609. IEEE, 2005.

[136] Edith Law and Luis von Ahn. Human computation. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 5(3):1–121, 2011.

[137] Florian Laws, Christian Scheible, and Hinrich Schütze. Active learning with amazon mechanical turk. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1546–1556. Association for Computational Linguistics, 2011.

[138] Yan Yan, Glenn M Fung, Rómer Rosales, and Jennifer G Dy. Active learning from crowds. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 1161–1168, 2011.

[139] Vamshi Ambati, Stephan Vogel, and Jaime G Carbonell. Active learning and crowd-sourcing for machine translation. In *LREC*, volume 1, page 2. Citeseer, 2010.

[140] John R Koza. *Genetic programming: on the programming of computers by means of natural selection*, volume 1. MIT press, 1992.

[141] Reuven Cohen and Shlomo Havlin. *Complex networks: structure, robustness and function*. Cambridge University Press, 2010.

[142] M Schmidt and H Lipson. Eureqa (version 0.98 beta)[software]. nutonian inc., cambridge, ma, 2014.

[143] Fang Wu, Dennis M Wilkinson, and Bernardo A Huberman. Feedback loops of attention in peer production. In *Computational Science and Engineering, 2009. CSE'09. International Conference on*, volume 4, pages 409–415. IEEE, 2009.

[144] Karl Raimund Popper, Karl Raimund Popper, and Karl Raimund Popper. Objective knowledge: An evolutionary approach. 1972.

[145] John Dewey. Experience and education. In *The Educational Forum*, volume 50, pages 241–252. Taylor & Francis, 1986.

[146] Daniel D Corkill. Collaborating software: Blackboard and multi-agent systems & the future. In *Proceedings of the International Lisp Conference*, volume 10, 2003.

[147] Gerwin Schalk, Dennis J McFarland, Thilo Hinterberger, Niels Birbaumer, and Jonathan R Wolpaw. Bci2000: a general-purpose brain-computer interface (bci) system. *Biomedical Engineering, IEEE Transactions on*, 51(6):1034–1043, 2004.

[148] Jonathan R Wolpaw, Dennis J McFarland, Gregory W Neat, and Catherine A Forneris. An eeg-based brain-computer interface for cursor control. *Electroencephalography and clinical neurophysiology*, 78(3):252–259, 1991.